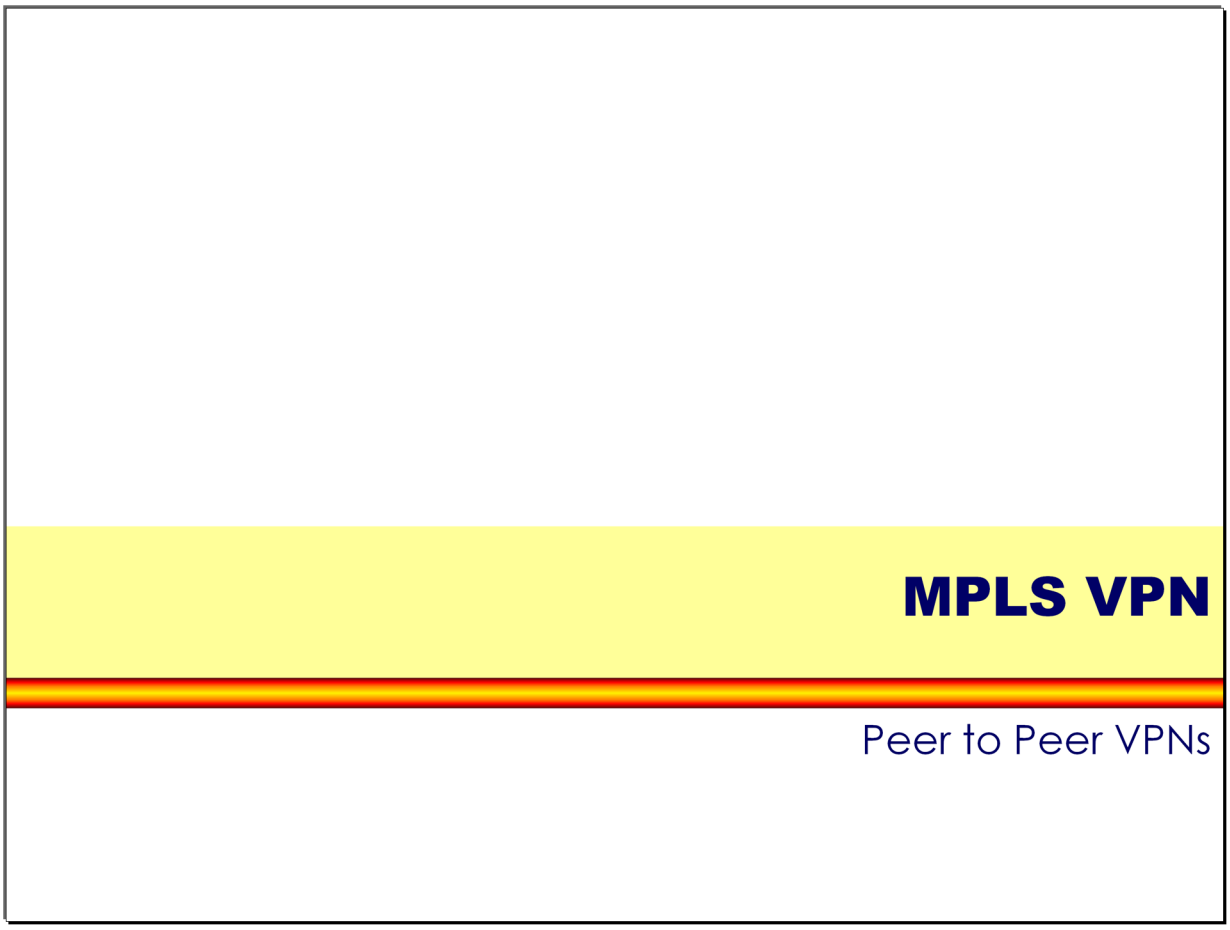
Appendix 4 - MPLS-VPN (v6.1)



Agenda

- **MP-BGP**
- **VPN Overview**
- **MPLS VPN Architecture**
- **MPLS VPN Basic VPNs**
- **MPLS VPN Complex VPNs**
- **MPLS VPN Configuration (Cisco)**
  - CE-PE OSPF Routing
  - CE-PE Static Routing
  - CE-PE RIP Routing
  - CE-PE External BGP Routing

Multiprotocol BGP 1

- **BGP-4 (RFC 1771) is capable of carrying routing information only for IPv4**
- **The only three pieces of information carried by BGP-4 that are IPv4 specific are**
  - the NEXT_HOP attribute (expressed as an IPv4 address),
  - the AGGREGATOR (contains an IPv4 address)
  - the NLRI (expressed as IPv4 address prefixes)
- **Multiprotocol Extensions to BGP-4**
  - RFC 2858
  - enable it to carry routing information for multiple network layer protocols (e.g., IPv6, IPX, etc...).

Multiprotocol BGP 2

- **To enable BGP-4 to support routing for multiple network layer protocols two things have to be added**
  - the ability to associate a particular network layer protocol with the next hop information
  - the ability to associate a particular network layer protocol with a NLRI
- **To identify individual network layer protocols**
  - Address Family Identifiers (AFI) are used
  - values defined in RFC 1700
    - RFC 1700 is historic, obsoleted by RFC 3232
    - RFC 3232 specifies a Online Database for ASSIGNED NUMBERS
      - www.iana.org

Address Family Numbers (RFC 1700)

Number	Description
0	Reserved
1	IP (IP version 4)
2	IP6 (IP version 6)
3	NSAP
4	HDLC (8-bit multidrop)
5	BBN 1822
6	802 (includes all 802 media plus Ethernet "canonical format")
7	E.163
8	E.164 (SMDS, Frame Relay, ATM)
9	F.69 (Telex)
10	X.121 (X.25, Frame Relay)
11	IPX
12	AppleTalk
13	Decnet IV
14	Banyan Vines
65535	Reserved

Multiprotocol BGP 4

- **Address Family Identifier (AFI) in MP-BGP**
  - this parameter is used to differentiate routing updates of different protocols carried across the same BGP session
  - it is a 16-bit value
- **MP-BGP uses an additional Sub-Address Family Identifier (SAFI)**
  - it is a 8-bit value
    - 1 NLRI used for unicast forwarding
    - 2 NLRI used for multicast forwarding
    - 3 NLRI used for both unicast and multicast forwarding
- **Usual notation AFI/SAFI (i.e. x/y)**
    - 1/1 IP version 4 unicast
    - 1/2 IP version 4 multicast
    - 1/128 VPN-IPv4 unicast (used for MPLS-VPN)

 MPLS-VPN v6.1 6

Multiprotocol BGP 3

- **Capability Advertisement Procedures are used**
  - by a BGP speaker that to determine whether the speaker could use multiprotocol extensions with a particular peer or not -> RFC 3392
  - done during BGP Open with Capabilities Optional Parameter (Parameter Type 2)

```
+------------------------------+
| Capability Code (1 octet)    |
+------------------------------+
| Capability Length (1 octet)  |
+------------------------------+
| Capability Value (variable)  |
+------------------------------+
```

    - Capability Code is unambiguously identifies individual capabilities. Capability Value is interpreted according to the value of the Capability Code field.

 MPLS-VPN v6.1 7

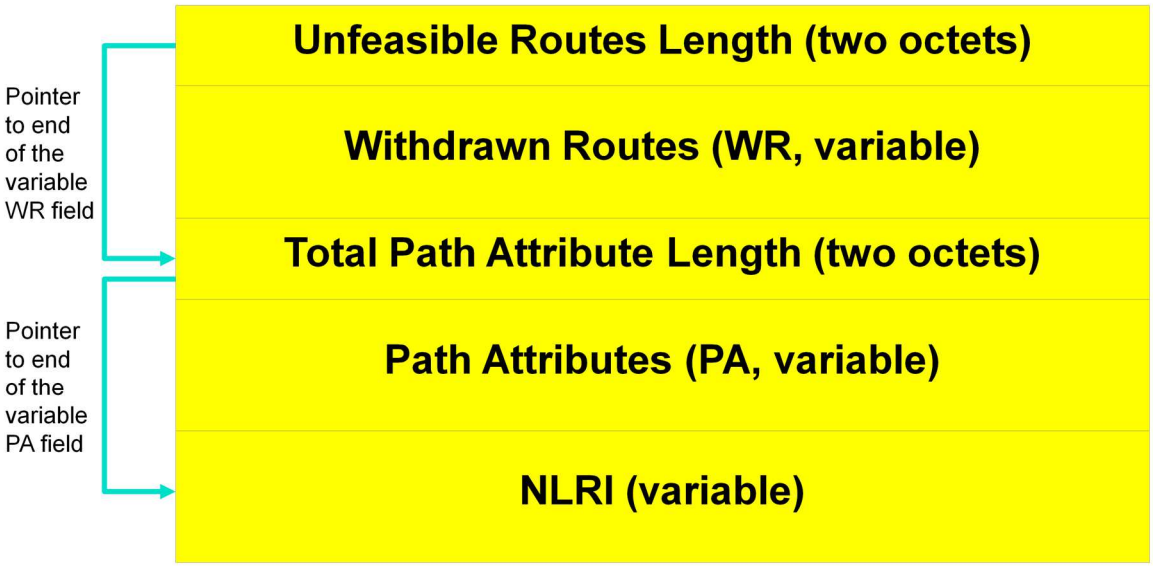
Capability Code values 1 through 63 are to be assigned by IANA using the "IETF Consensus" policy defined in RFC 2434. Capability Code values 64 through 127 are to be assigned by IANA, using the "First Come First Served" policy defined in RFC 2434. Capability Code values 128 through 255 are for "Private Use" as defined in RFC 2434.

Multiprotocol BGP 4

- **Two new attributes**
  - Multiprotocol Reachable NLRI (MP_REACH_NLRI)
  - Multiprotocol Unreachable NLRI (MP_UNREACH_NLRI)
- **MP_REACH_NLRI is used**
  - to carry the set of reachable destinations together with the next hop information to be used for forwarding to these destinations
- **MP_UNREACH_NLRI is used**
  - to carry the set of unreachable destinations
- **Both of these attributes**
  - are optional and non-transitive

Appendix 4 - MPLS-VPN (v6.1)

BGP Update Message Format for IPv4



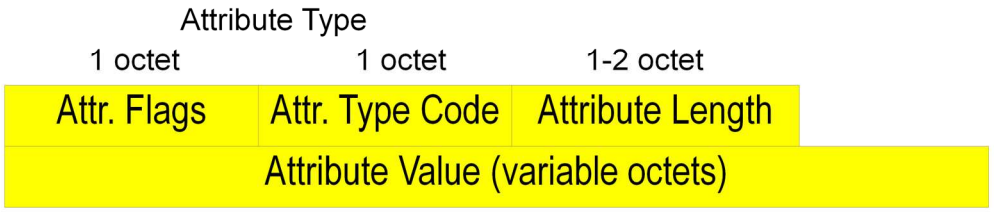
 MPLS-VPN v6.1 9

BGP Update Message Details for IPv4

- **NLRI**
  - 2-tuples of (length, prefix)
    - length = number of masking bits (1 octet)
    - prefix = IP address prefix (1 - 4 octets)
    - note: prefix field contains only necessary bits to completely specify the IP address followed by enough trailing bits to make the end of the field fall on an octet boundary
- **path attributes are composed of**
  - triples of (type, length, value) -> TLV notation
    - attribute type (two octets)
      - 8 bit attribute flags, 8 bit attribute type code
    - attribute length (one or two octets)
      - signaled by attribute flag-bit nr.4
    - attribute value (variable length)
      - content depends on meaning signaled by attribute type code

Appendix 4 - MPLS-VPN (v6.1)

IPv4 Path Attribute Format / NLRI Format



Path Attribute Format

NLRI

Appendix 4 - MPLS-VPN (v6.1)

VPN-IPv4 BGP Update with MP_Reach_NLRI

1 octet	1 octet	1 octet
Attr. Flags	Type Code = 14	Attribute Length
AFI= 1		SAFI = 128
Length of NHA		
Next Hop Address (NHA, 1- 4 octets)		

Path Attribute MP_Reach_NLRI

1 octet
Length = 120
Label (3 octets = 24 bits)
Route Distinguisher (8 octets = 64 bits)
IPv4 address (4 octets = 32 bits)

NLRI for VPN-IPv4

 MPLS-VPN v6.1 12

Format of Attribute-Type

- **8 bit attribute flags**
  - 1. bit (MSB)
    - optional (1) or well-known (0)
  - 2. bit
    - transitive (1) or non-transitive (0)
    - only for optional; set to 1 for well-known
  - 3. bit
    - partial (1) or complete (0)
    - set to 0 for well-known and optional non-transitive
  - 4. bit
    - two octet (1) or one octet (0) attribute length field
- **8 bit attribute type code**
  - values 1 - 16 currently defined

Classification of Attributes 1

- **well-known**
  - must be recognized by all BGP implementations
- **well-known mandatory**
  - must be included in every Update message
    - Origin, AS_Path, Next_Hop
- **well-known discretionary**
  - may or may not be included in every Update message
    - Local_Preference, Atomic_Aggregate
- **all well-known attributes must be passed along to other BGP peers**
  - some will be updated properly first, if necessary

Classification of Attributes 2

- **optional**
  - it is not required or expected that all BGP implementation support all optional attributes
  - may be added by the originator or any AS along the path
  - paths are accepted regardless whether the BGP peer understands an optional attribute or not
- **handling of recognized optional attributes**
  - propagation of attribute depends on meaning of the attribute
  - propagation of attribute is not constrained by transitive bit of attribute flags
    - but depends on the meaning of the attribute

Classification of Attributes 3

- **handling of unrecognized optional attribute**
  - propagation of attribute depends on transitive bit of attribute flags
  - transitive
    - paths are accepted (attribute is ignored) and attribute remains unchanged when path is passed along to other peers
    - attribute is marked as partial (bit 3 of attribute flags)
    - example: Community
  - non-transitive
    - paths are accepted, attribute is quietly ignored and discarded when path is passed along to other peers
    - example: Multi_Exit_Discriminator

Currently Defined Attributes 1

- **Basic attributes**
  - defined in RFC 1771 (Draft Standard)
  - Origin
    - well-known mandatory; type 1
  - AS_Path
    - well-known mandatory; type 2
  - Next_Hop
    - well-known mandatory; type 3
  - Multi_Exit_Discriminator MED
    - optional non-transitive; type 4
  - Local_Preference
    - well-known discretionary; type 5

Currently Defined Attributes 2

- **Basic attributes (cont.)**
  - Atomic_Aggregate
    - well-known discretionary; type 6
  - Aggregator
    - optional transitive; type 7
  - these are the attributes that you can rely on in a multi-vendor environment

Currently Defined Attributes 3

- **Advanced attributes**
  - Community
    - optional transitive; type 8
    - defined in RFC 1997 (Proposed Standard)
  - Originator_ID
    - optional non-transitive; type 9
    - defined in RFC 1966 (Experimental) and RFC 2796 (Proposed Standard) -> Route Reflector
  - Cluster_List
    - optional non-transitive; type 10
    - defined in RFC 1966 (Experimental) and RFC 2796 (Proposed Standard) -> Route Reflector

Currently Defined Attributes 4

- **Advanced attributes (cont.)**
  - Multiprotocol Reachable NLRI
    - MP_REACH_NLRI
    - optional non-transitive; type 14
    - defined in RFC 2858 (Proposed Standard) -> Multiprotocol Extensions
  - Multiprotocol Unreachable NLRI
    - MP_UNREACH_NLRI
    - optional non-transitive; type 15
    - defined in RFC 2858 (Proposed Standard) -> Multiprotocol Extensions
  - in a multi-vendor environment carefully check implementation details

Community Attribute Review 1

- **optional transitive attribute**
- **community is a group of destinations that share a common property**
  - group of networks which should be handled by a foreign AS in a certain way
  - community is not restricted to one network or one AS
- **community attributes are used**
  - to simplify routing policy based on logical properties rather than IP prefix or AS number (= physical location)
  - to tag routes to ensure consistent filtering or route-selection policy

Community Attribute Review 2

- **32 bit values (range 0 - 4.294.967.200)**
- **well-known communities**
  - 0xFFFFFF01 … No_Export
  - 0xFFFFFF02 … No_Advertise
- **private communities**
  - value range 0x00010000 to 0xFFFEFFFF
  - common practice for using private communities:
    - high order 16 bit: number of AS
      - which is responsible for defining the meaning of the community
    - low order 16 bit: definition of meaning
      - might have only local significance within the defining AS

BGP Draft Attributes 1

- **BGP Extended Communities Attribute**
  - consists of a set of "extended communities”
    - optional transitive; type 16
    - defined in draft-ietf-idr-bgp-ext-communities-07.txt
  - two important enhancements over the existing BGP Community Attribute:
    - it provides an extended range, ensuring that communities can be assigned for a plethora of uses, without fear of overlap.
    - the addition of a type field provides structure for the community space.
  - Important for MPLS_VPN
    - **Route Target Community**
    - **Route Origin Community**

 MPLS-VPN v6.1 23

plethora: Unmenge, Fülle

BGP Draft Attributes 2

- **Route Target:**
  - The Route Target Community identifies one or more routers that may receive a set of routes (that carry this Community) carried by BGP. This is transitive across the Autonomous system boundary.
  - It really identifies only a set of sites which will be able to use the route, without prejudice to whether those sites constitute what might intuitively be called a VPN.
- **Route Origin:**
  - The Route Origin Community identifies one or more routers that inject a set of routes (that carry this Community) into BGP. This is transitive across the Autonomous system boundary.

BGP Draft Attributes 3

- **Route Target and Router Origin**
  - type: 2 octets (extended form of this attribute)
    - high octet -> 00, 01, 02 -> defines the structure of the value field
    - low octet -> defines the actual type
  - value: 6 octets
- **Route Target:**
  - high octet type: 0x00 or 0x01 or 0x02
  - low octet type: 0x02
- **Route Origin:**
  - high octet type: 0x00 or 0x01 or 0x02
  - low octet type: 0x03

BGP Draft Attributes 4

- **Structure of value field based on high octet part of type**
  - 0x00:
    - 2 octets Global Administrator Field (IANA assigned AS #)
    - 4 octets Local Administrator Field (actual value of given type contained in low octet part of type)
  - 0x01:
    - 4 octets Global Administrator Field (IP address assigned by IANA)
    - 2 octets Local Administrator Field
  - 0x02:
    - 4 octets Global Administrator Field (IANA assigned 4 octet AS #)
    - 2 octets Local Administrator Field

Agenda

- **MP-BGP**
- **VPN Overview**
- **MPLS VPN Architecture**
- **MPLS VPN Basic VPNs**
- **MPLS VPN Complex VPNs**
- **MPLS VPN Configuration (Cisco)**
  - CE-PE OSPF Routing
  - CE-PE Static Routing
  - CE-PE RIP Routing
  - CE-PE External BGP Routing

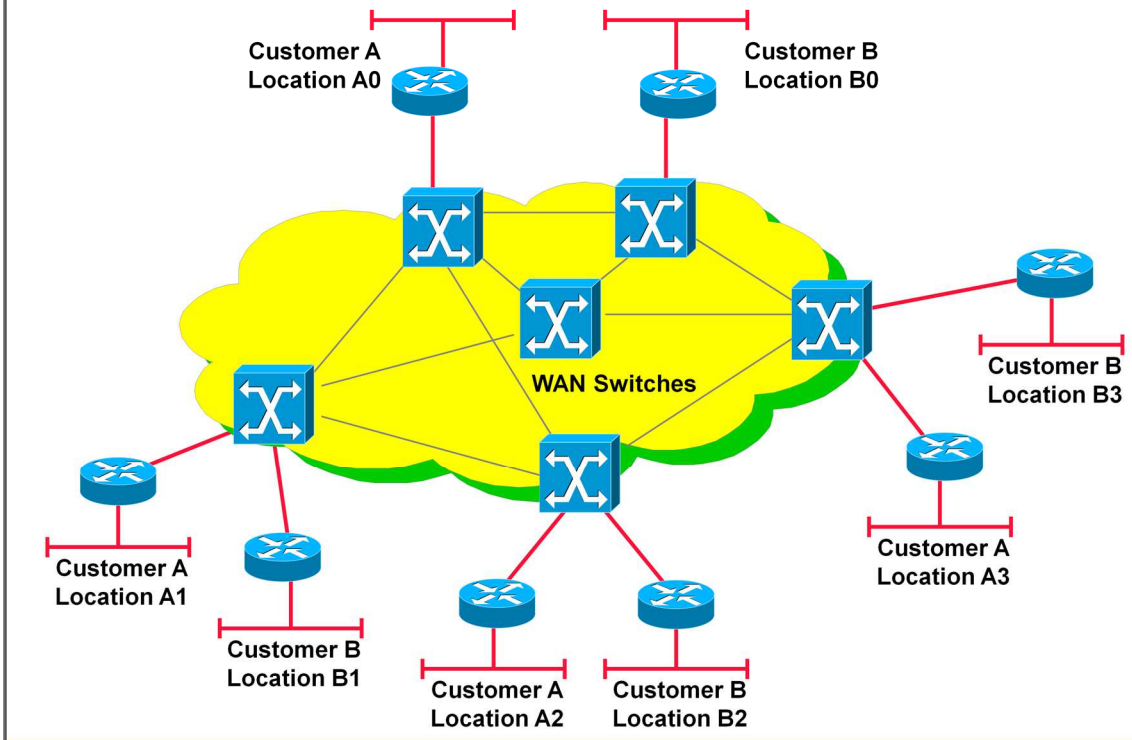
Classical VPNs

- X.25, Frame Relay or ATM in the core
- dedicated physical switch ports for every customers CPE
  - router, bridge, computer
- customer traffic separation in the core done by concept of virtual circuit
  - PVC service
    - management overhead
  - SVC service with closed user group feature
    - signaling overhead
- separation of customers inherent to virtual circuit technique
- privacy is aspect of customer
  - in most cases overlooked

VPNs based on Overlay Model

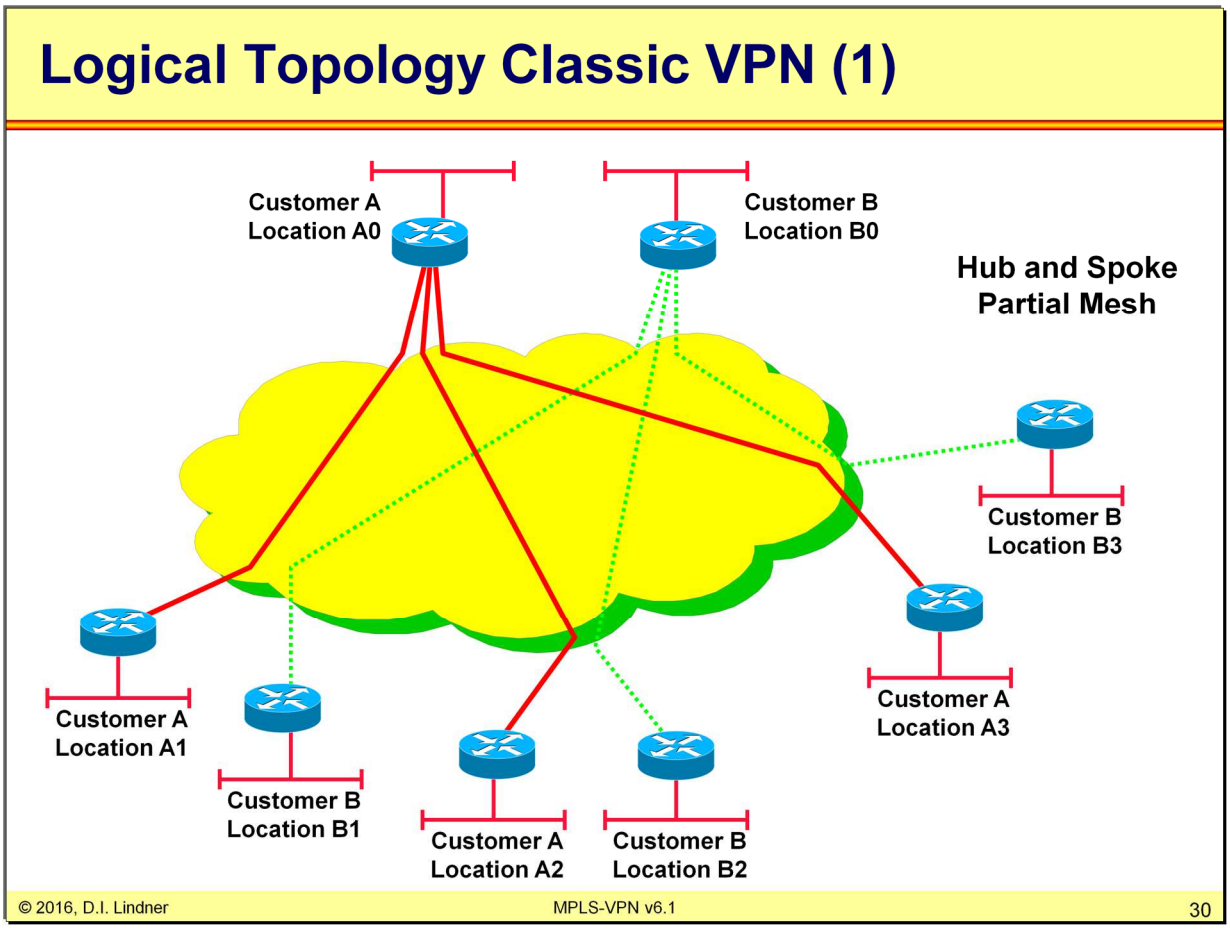
Appendix 4 - MPLS-VPN (v6.1)

Physical Topology of Classical VPN



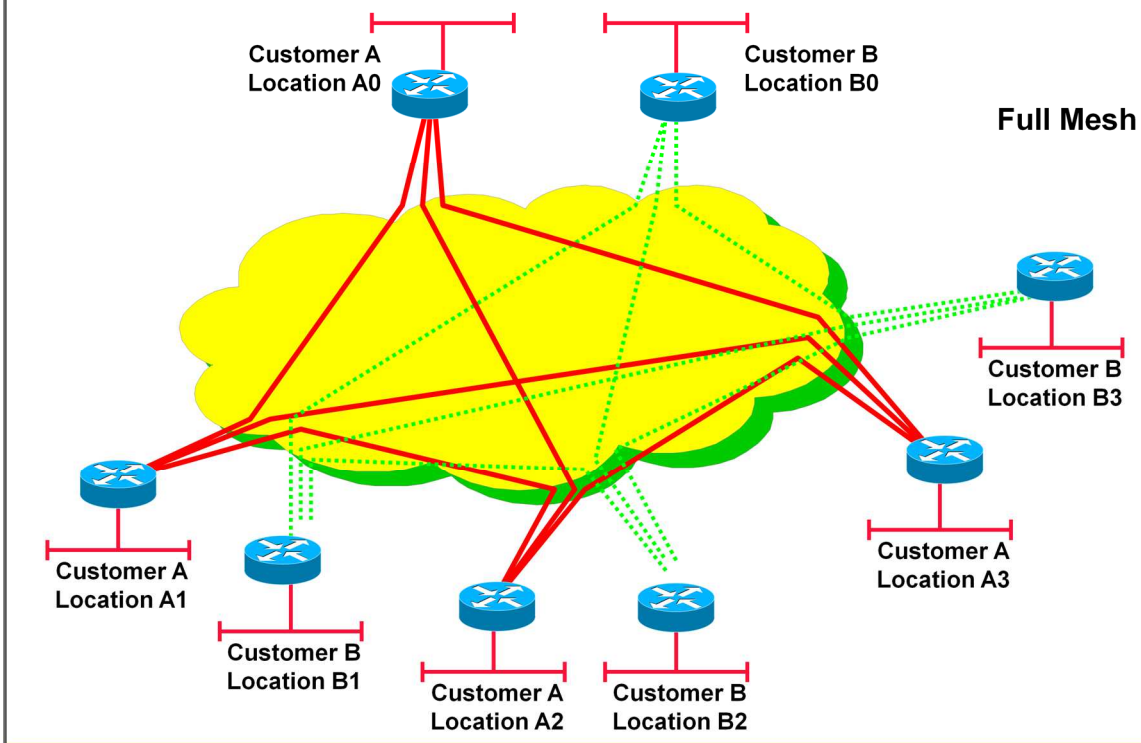
MPLS-VPN v6.1
29

Appendix 4 - MPLS-VPN (v6.1)



Appendix 4 - MPLS-VPN (v6.1)

Logical Topology Classic VPN (2)



Virtual Private Networks based on IP

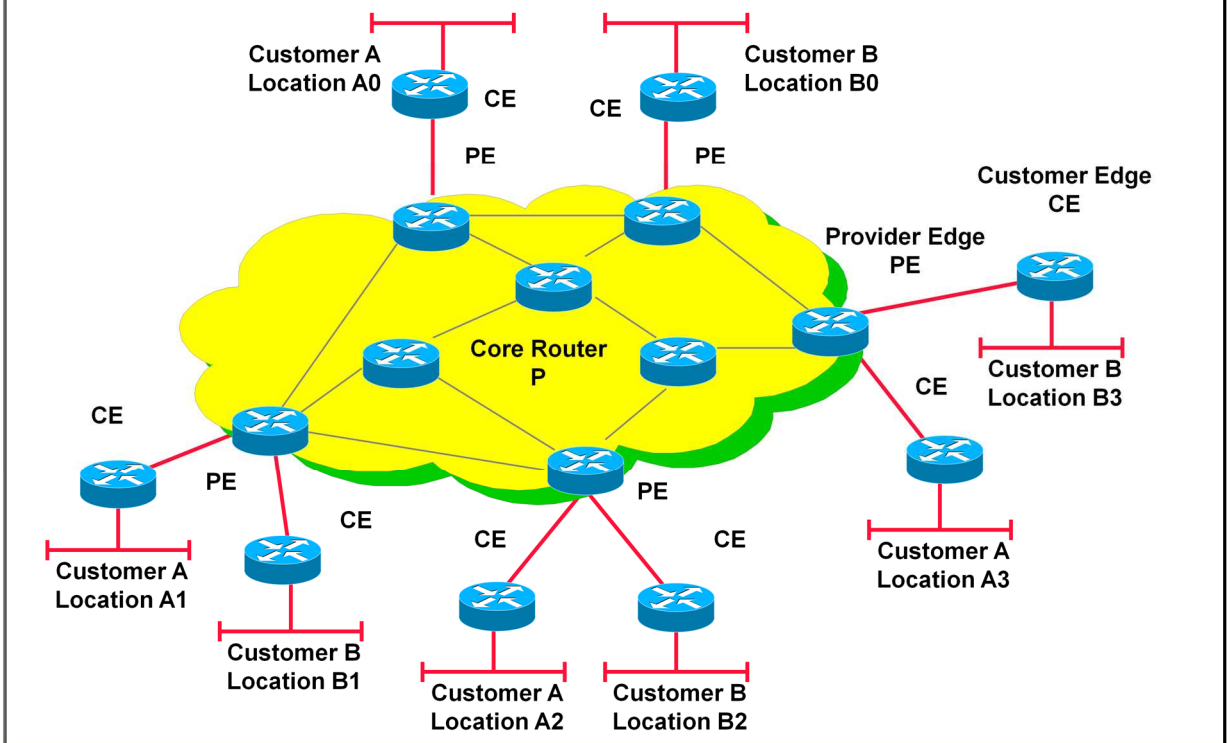
- single technology end-to-end
  - IP forwarding and IP routing
- no WAN switches in the core
  - based on different technology (X.25, FR or ATM)
  - administered by different management techniques
- but accounting and quality of service just coming in the IP world
  - X.25, FR and ATM have it already
- often private means cases control over separation but not privacy
  - data are seen in clear-text in the core
  - encryption techniques can solve this problem
  - but encryption means must be in the hand of the customer

VPNs based on Peer Model

 MPLS-VPN v6.1 32

Appendix 4 - MPLS-VPN (v6.1)

Physical Topology IP VPN



 MPLS-VPN v6.1 33

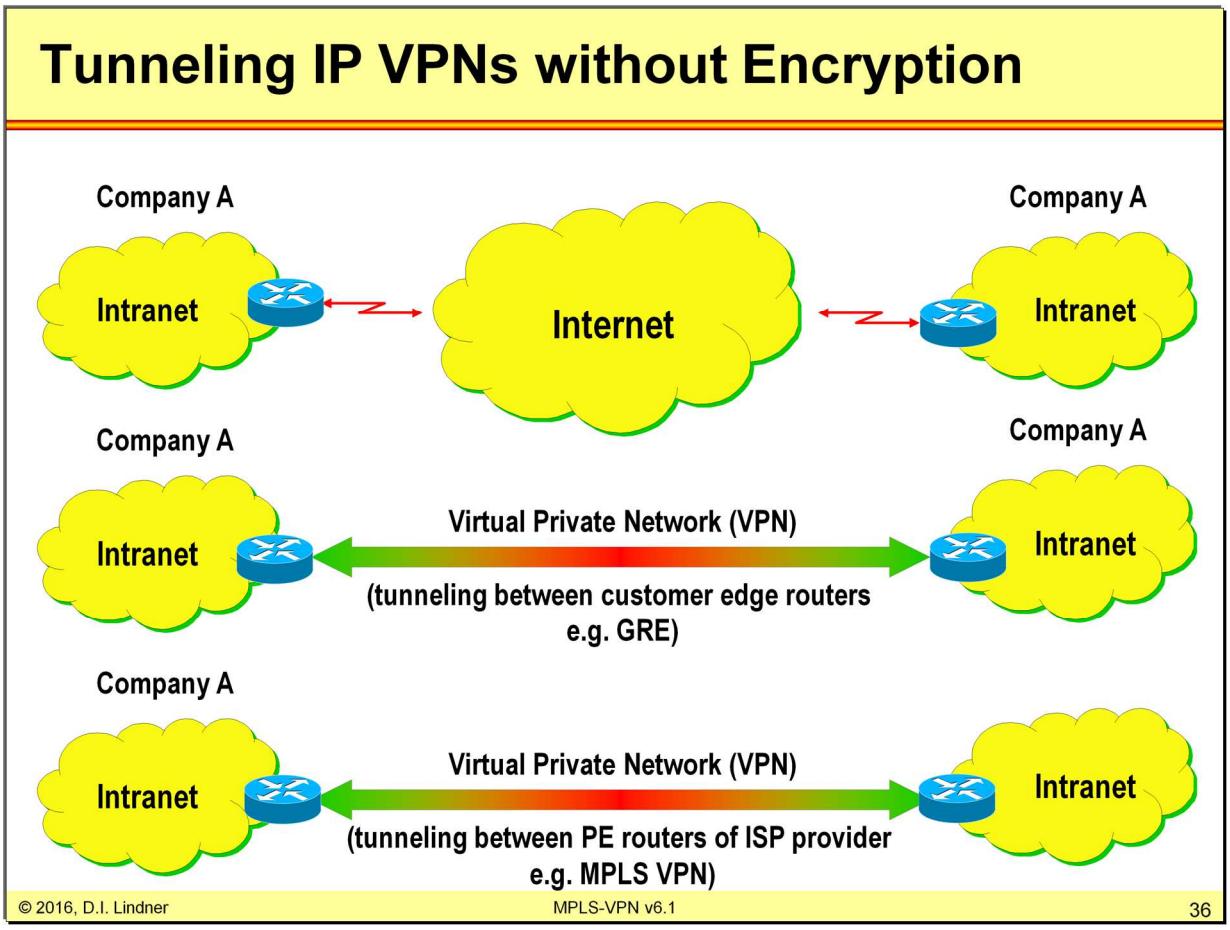
Possible Solutions for IP VPNs

- **IP addresses of customers non overlapping**
  - filtering and policy routing techniques can be used in order to guarantee separation of IP traffic
    - exact technique depends on who manages routes at the customer site
- **IP addresses of customers overlapping**
  - tunneling techniques must be used in order to guarantee separation of IP traffic
    - GRE
    - L2F, PPTP, L2TP
    - MPLS-VPN
- **If privacy is a topic**
  - encryption techniques must be used
    - SSL/TLS, IPsec

Tunneling Solutions for IP VPNs

- **Tunneling techniques are used in order to guarantee separation of IP traffic**
  - IP in IP Tunneling or GRE (Generic Routing Encapsulations)
    - Bad performance on PE router
  - PPTP or L2TP for LAN to LAN interconnection
    - Originally designed for PPP Dial-up connections
    - LAN – LAN is just a special case
  - MPLS-VPN
    - Best performance on PE router
- **In all these cases**
  - Privacy still an aspect of the customer

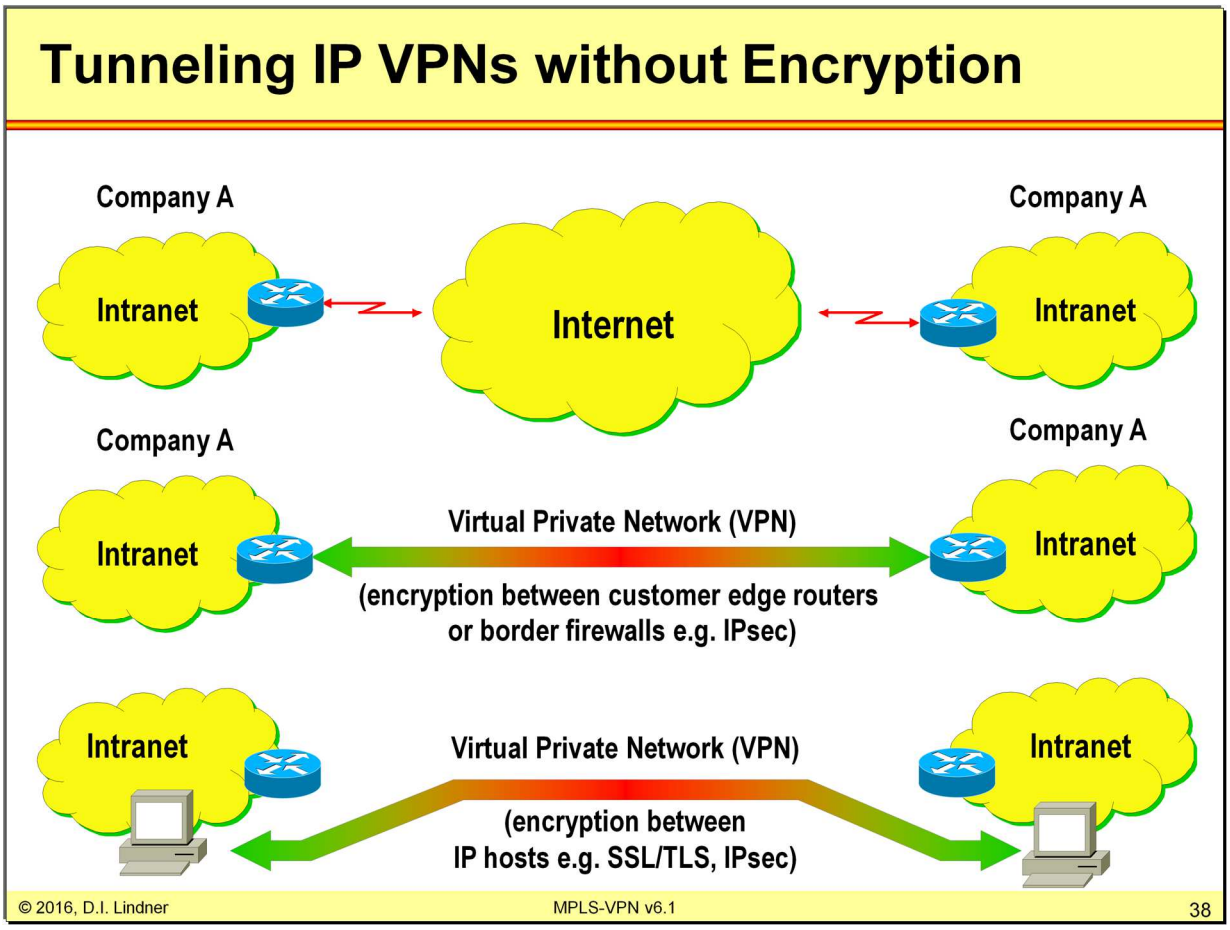
Appendix 4 - MPLS-VPN (v6.1)



Encryption Solutions for IP VPNs

- **If privacy is a topic tunneling techniques with encryption are used in order to hide IP traffic**
  - SSL (secure socket layer)
    - Usually end-to-end
    - Between TCP and Application Layer
  - IPsec
    - Could be end-to-end
    - Could be between special network components (e.g. firewalls, VPN concentrators) only
    - Between IP and TCP/UDP Layer
  - PPTP and L2TP Tunnels
    - With encryption turned on via PPP option

Appendix 4 - MPLS-VPN (v6.1)

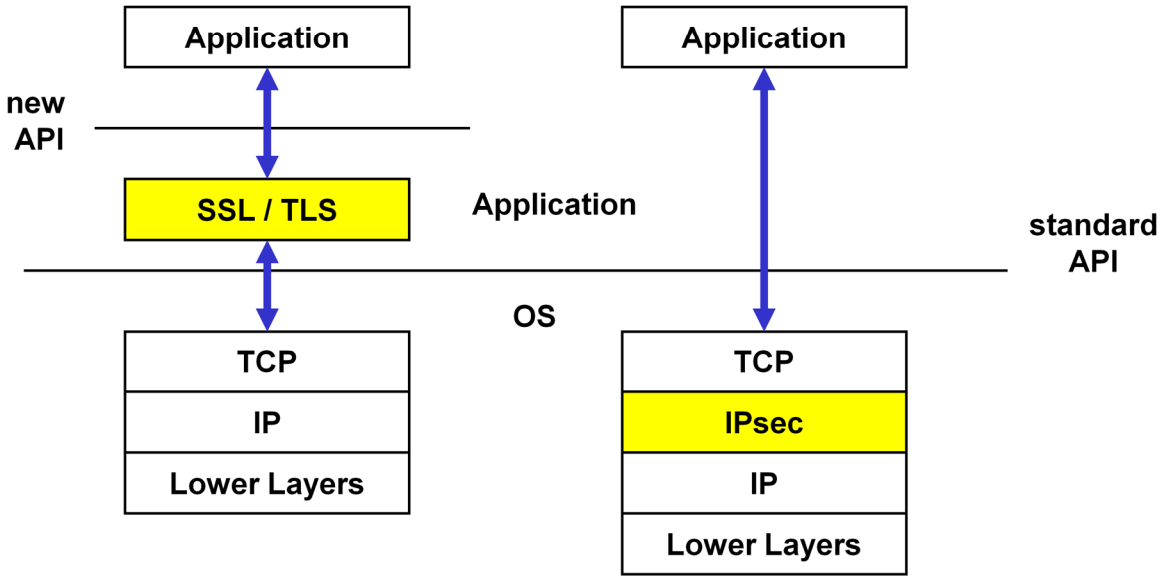


Appendix 4 - MPLS-VPN (v6.1)

SSL/TLS versus IPsec

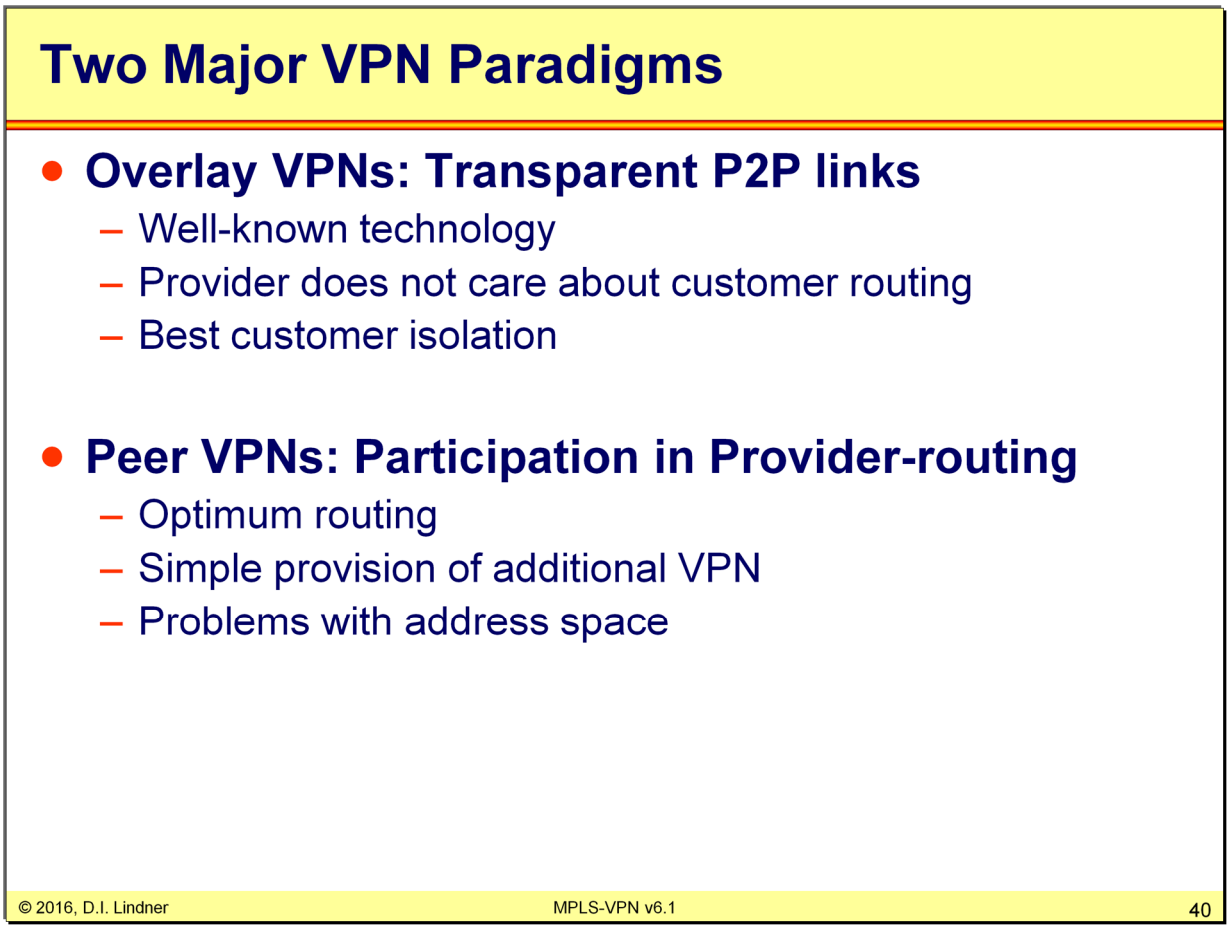
Application must be aware of new application programming interface

Application can use standard application programming interface



 MPLS-VPN v6.1 39

Appendix 4 - MPLS-VPN (v6.1)



VPN services can be offered based on two major paradigms:

Overlay VPNs requires service providers to provide virtual point-to-point links between customer sites. The service provider does not see customer routes and is responsible only for providing point-to-point transport of customer data. All routing protocols run directly between customer routers.

Layer 1 solutions: Classical TDM technologies such as E1, ISDN, SONET/SDH.

Layer 2 solutions: FR, ATM, X.25.

Layer 3 solutions: IPsec, GRE whereas access (dialup) environments use L2TP, PPTP or L2F.

Peer-to-Peer VPNs requires service providers to participate in customer routing.

The isolation of the customers is realized via packet filters on PE routers at the PE-CE interfaces.

Another alternative is to implement controlled route distribution where each customer has a dedicated PE router which only knows about this customer's routes.

Peer VPNs allow a much simpler provision of additional VPNs because only the sites are provisioned, not the links between them.

Note: All customers share the same (provider-assigned or public) address space.

MPLS VPN – Best of Both Worlds

- **Combines VPN Overlay model with VPN Peer model**
- **PE routers allow route isolation**
  - By using Virtual Routing and Forwarding Tables (VRF) for differentiating routes from the customers
  - Allows overlapping address spaces
- **PE routers participate in P-routing**
  - Hence optimum routing between sites
  - Label Switches Paths are used within the core network
  - Easy provisioning (sites only)
- **Overlapping VPNs possible**
  - By a simple (?) attribute syntax

 MPLS-VPN v6.1 41

The MPLS VPN solution combines the best of both worlds (overlapping and peer VPN).

Here the PE routers participate in C-routing which allows for easy provisioning and optimum site-connections. But the core routers do not need to carry much routing information. Only the PE routers must have some power.

Site isolation is provided by Virtual Routing and Forwarding Tables (VRFs) which are explained soon. This method allows for overlapping address space or overlapping VPNs (but not both together).

The main task is to specify which routes should be imported into which VRF. This is accomplished by special attributes during the configuration. The principle is easy (as you will see) but the attribute-syntax looks...strange (as you will see).

Appendix 4 - MPLS-VPN (v6.1)

MPLS VPN – Principles

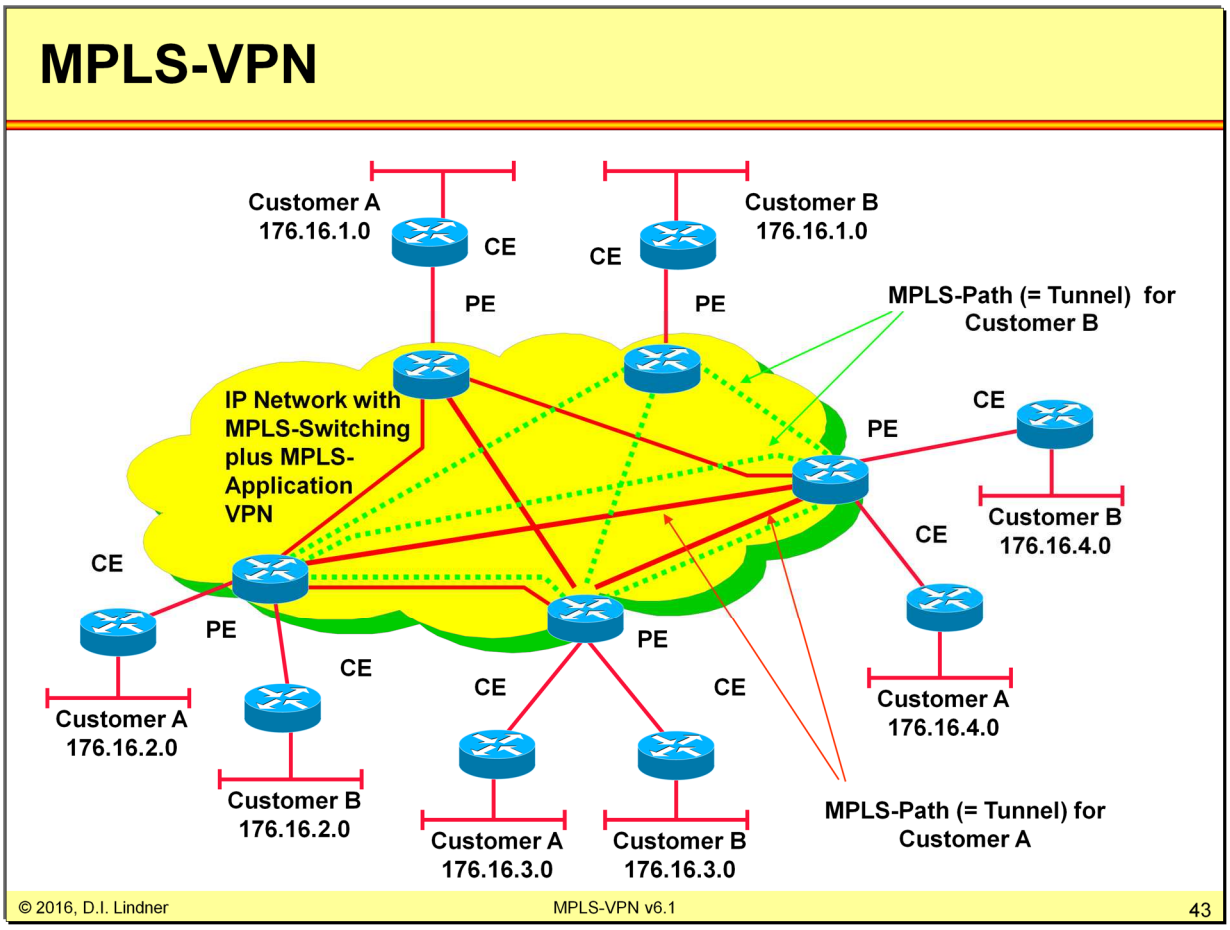
- **Requires MPLS Transport within the core**
  - Using the label stack feature of MPLS
- **Requires MP-BGP among PE routers**
  - Supports IPv4/v6, VPN-IPv4, multicast
  - Default behavior: BGP-4
- **Requires VPN-IPv4 96 bit addresses**
  - 64 bit Route Distinguisher (RD)
  - 32 bit IP address
- **Every PE router uses one VRF for each VPN**
  - Virtual Routing and Forwarding Table (VRF)

 MPLS-VPN v6.1 42

For MPLS VPN services its mandatory to have an properly working MPLS Transport system already in place. Furthermore MP-BGP needs to be set up to allow the exchange of VPNV4 updates and VPN Label information.

A VPNV4 address is made up of a 64 bit Route Distinguisher (RD) and a 32 bit IPV4 address. This VPNV4 address is needed to allow overlapping address spaces inside different VPNs. Every PE router holds different VRFs which holds address information for one or more VPNs, depending whether simple VPNs or overlapping VPNs are in use.

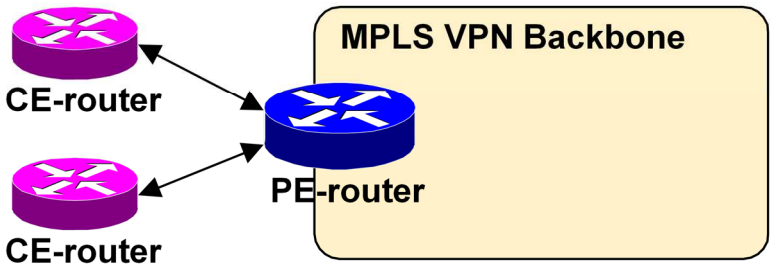
Appendix 4 - MPLS-VPN (v6.1)



Agenda

- **MP-BGP**
- **VPN Overview**
- **MPLS VPN Architecture**
- **MPLS VPN Basic VPNs**
- **MPLS VPN Complex VPNs**
- **MPLS VPN Configuration (Cisco)**
  - CE-PE OSPF Routing
  - CE-PE Static Routing
  - CE-PE RIP Routing
  - CE-PE External BGP Routing

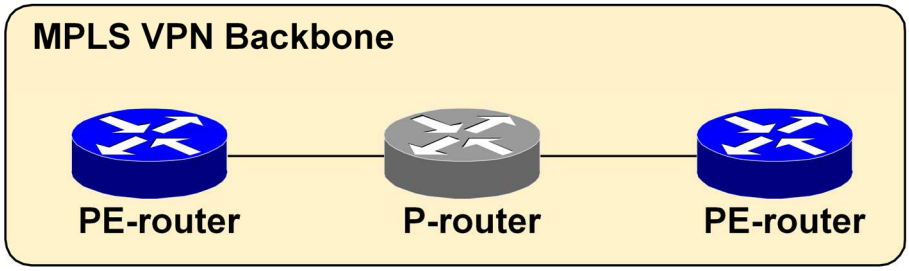
CE-Router Perspective



- **CE (Customer Edge) - routers run standard IP routing software and exchange routing updates with the PE-router**
  - EBGP, OSPF, RIPv2 or static routes are supported
- **PE (Provider Edge) - router appears as just another router in the customer's network**

Appendix 4 - MPLS-VPN (v6.1)

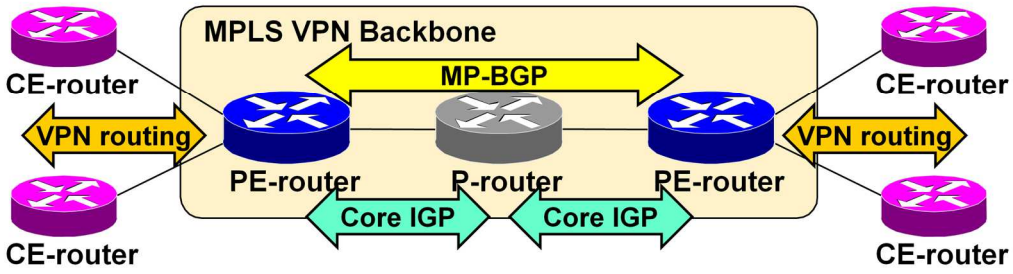
P-Router Perspective



- P (Provider) - routers do not participate in MPLS VPN routing and do not carry VPN (customer) routes
- P - routers run backbone IGP like OSPF or IS-IS with the PE-routers

Appendix 4 - MPLS-VPN (v6.1)

PE-Router Perspective



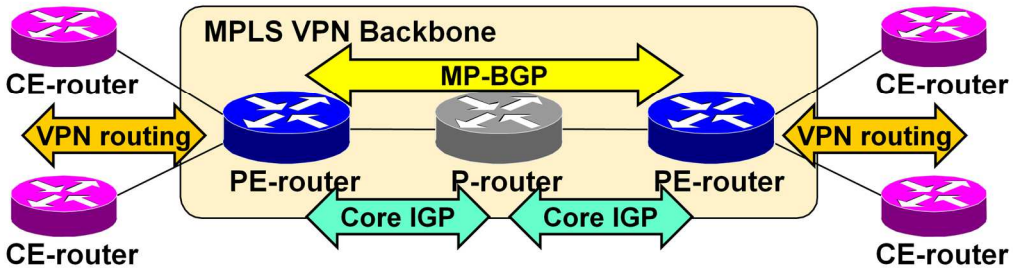
PE-routers contain a number of routing tables:

- Global routing table that contains core routes (filled with core IGP)
- Virtual Routing and Forwarding (VRF) tables for sets of sites with identical routing requirements
- VRF's are filled with information from CE-routers and MP-BGP information from other PE-routers

 MPLS-VPN v6.1 47

Appendix 4 - MPLS-VPN (v6.1)

PE-Router Perspective

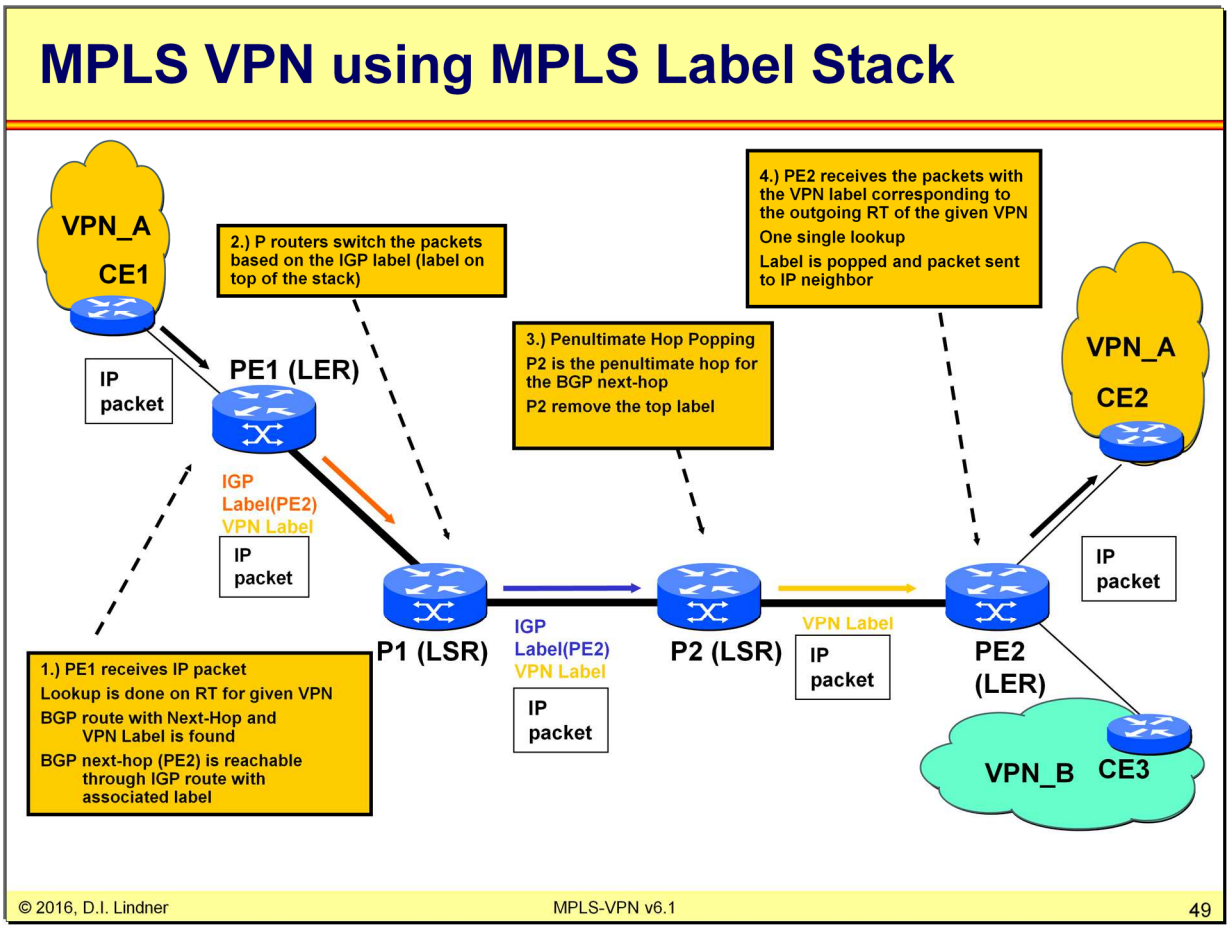


PE-routers:

- Exchange VPN routes with CE-routers via per-VPN routing protocols
- Exchange core routes with P-routers and PE-routers via core IGP
- Exchange VPN-IPv4 routes with other PE-routers via Internal MP-BGP sessions

 MPLS-VPN v6.1 48

Appendix 4 - MPLS-VPN (v6.1)

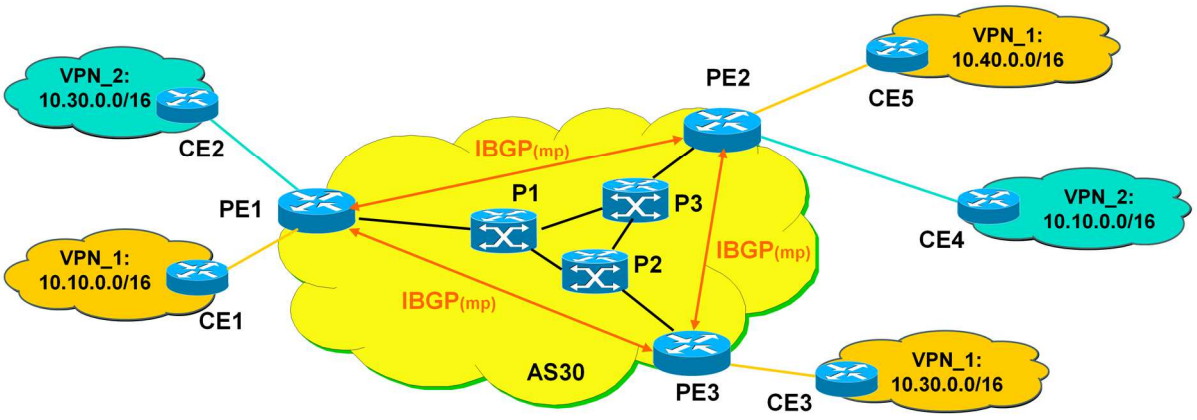


Agenda

- **MP-BGP**
- **VPN Overview**
- **MPLS VPN Architecture**
- **MPLS VPN Basic VPNs**
- **MPLS VPN Complex VPNs**
- **MPLS VPN Configuration (Cisco)**
  - CE-PE OSPF Routing
  - CE-PE Static Routing
  - CE-PE RIP Routing
  - CE-PE External BGP Routing

Appendix 4 - MPLS-VPN (v6.1)

VPN MPLS Architecture 1

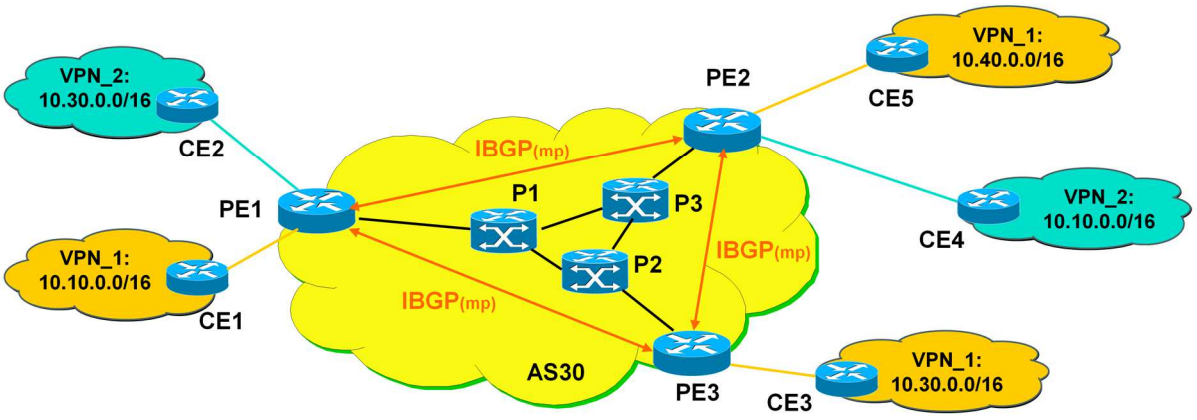


- **Service provider offers MPLS-VPN based on internal MPLS switching infrastructure**
  - PE … provider edge MPLS edge router, P … provider internal MPLS core router
  - CE … customer edge, conventional, IP router
- **MPLS-VPN requires full mesh of internal multiprotocol (mp) BGP sessions**
  - Could lead to a scalability problem in large environments
- **Customers receiving an IP VPN service**
  - Customer “Orange” and “Green”
  - Each customer has its own IP address space (VPN-1 or VPN_2) which is separated by MPLS-VPN
  - Address may overlap

 MPLS-VPN v6.1 51

Appendix 4 - MPLS-VPN (v6.1)

VPN MPLS Architecture 2

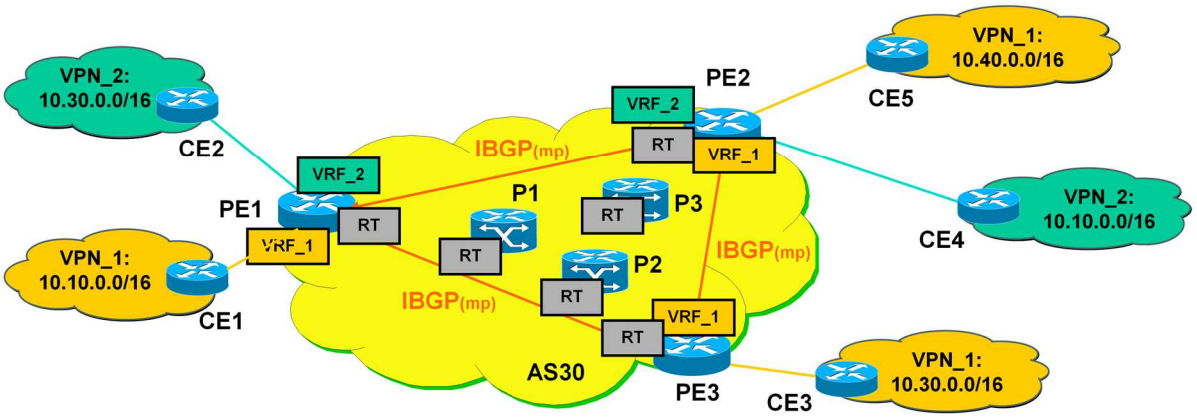


- Provider routers P (LSRs) are in the core of the MPLS cloud
- Provider Edge routers PE (LER) use MPLS within the core and plain IP with CE routers
- PE routers are fully meshed concerning Internal MP-BGP Sessions
- P and PE routers share a common IGP (e.g. OSPF or IS-IS)
- Customer Edge CE routers connect customer sites to provider

 MPLS-VPN v6.1 52

Appendix 4 - MPLS-VPN (v6.1)

VPN MPLS Architecture 3

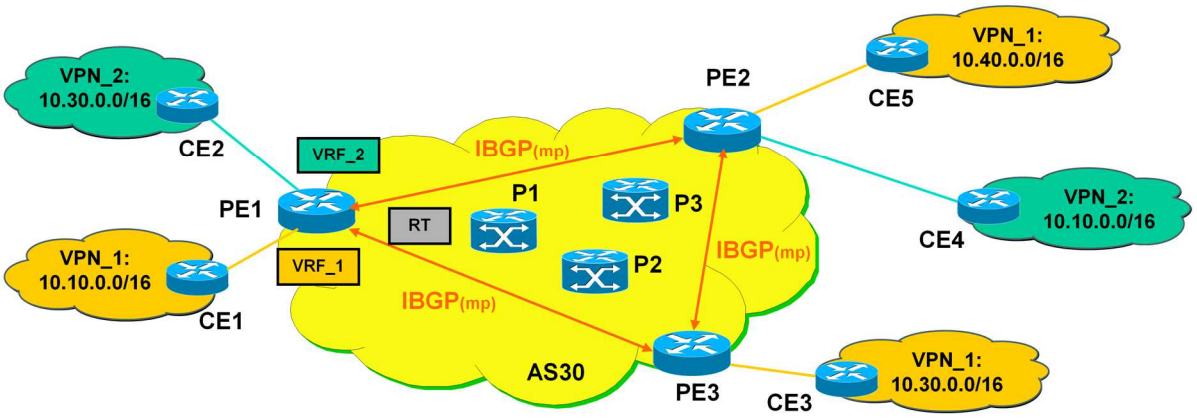


- PE router
  - maintains a separate routing table VRF per customer site
    - VRF (VPN Routing and Forwarding) Table
  - holds global routing table RT for communication within MPLS cloud
    - maintained by IGP
  - forwarding within MPLS cloud is based on labels
    - distributed by LDP

 MPLS-VPN v6.1 53

Appendix 4 - MPLS-VPN (v6.1)

VPN MPLS Architecture 4

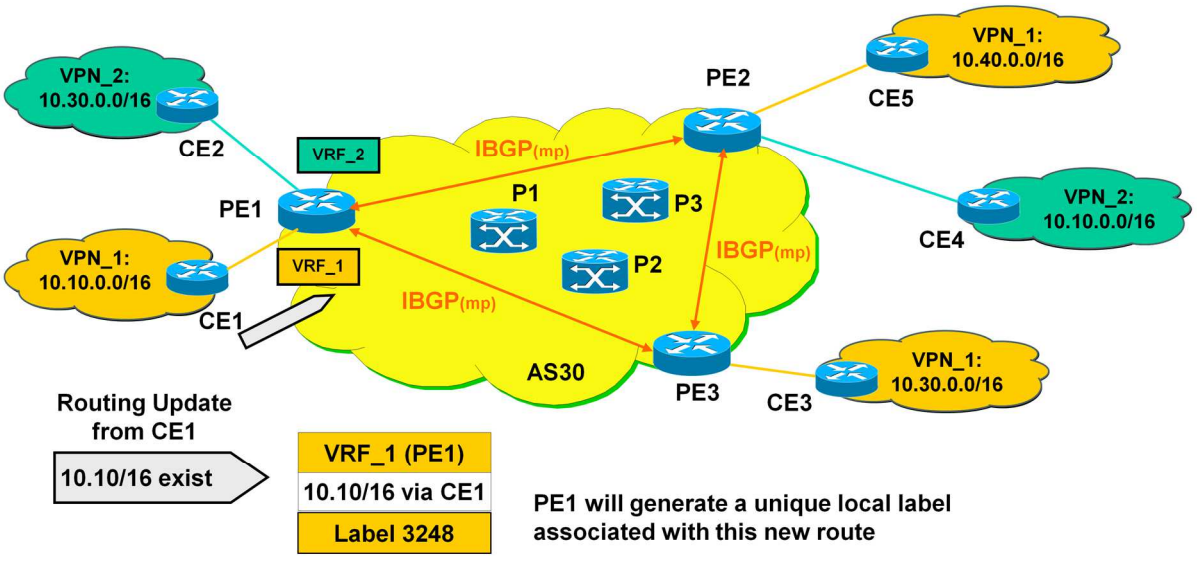


- VRF table
  - contains Net-IDs received from corresponding CE site
    - via RIPv2, OSPF, External BGP session or static routes
  - contains NET-IDs received from other PE routers
    - via Internal MP-BGP Sessions received as VPN-IPv4 addresses
  - hence overlapping addresses are no problem

 MPLS-VPN v6.1 54

Appendix 4 - MPLS-VPN (v6.1)

New Network 10.10.0.0/16 at CE1 1



Routing Update from CE1

10.10/16 exist

VRF_1 (PE1)
10.10/16 via CE1
Label 3248

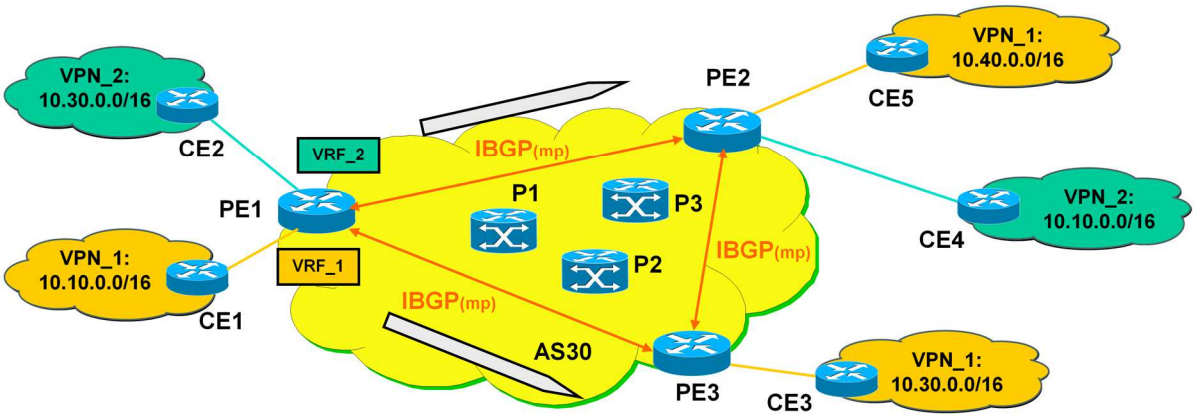
PE1 will generate a unique local label associated with this new route

- Routing Update will install a new route in the corresponding VRF table of PE1 and hence the new route must be advertised to all other PEs via Internal MP-BGP
  - as VPN-IPv4 address

 MPLS-VPN v6.1 55

Appendix 4 - MPLS-VPN (v6.1)

Advertise Network 10.10.0.0/16 to PE's 2



MP-BGP uses:

MP_Reach_NLRI attribute
Next-Hop
VPN-IPv4_NLRI
RD=Route Distinguisher
Net
Label
Extended Community attr.
RT = Route Target

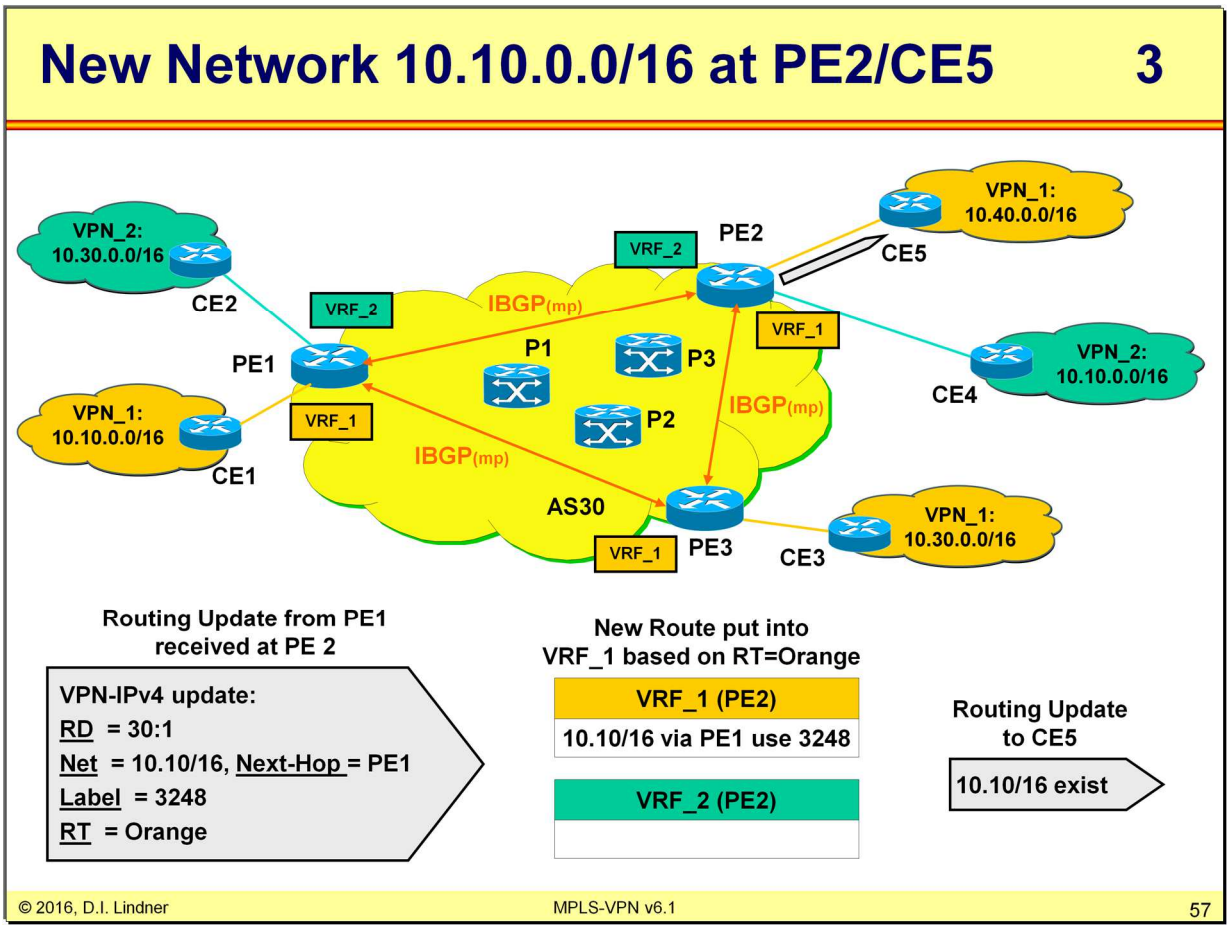
Routing Update from PE1 via Internal MP-BGP to all other PE´s

AS30 VPN #1

VPN-IPv4 update:
RD (ID to uniquely distinguished Net from other nets) = 30:1
Net = 10.10/16, Next-Hop = PE1
Label that should be used to reach this Net = 3248
RT (Hint to which VRF´s this Net should be imported) = Orange

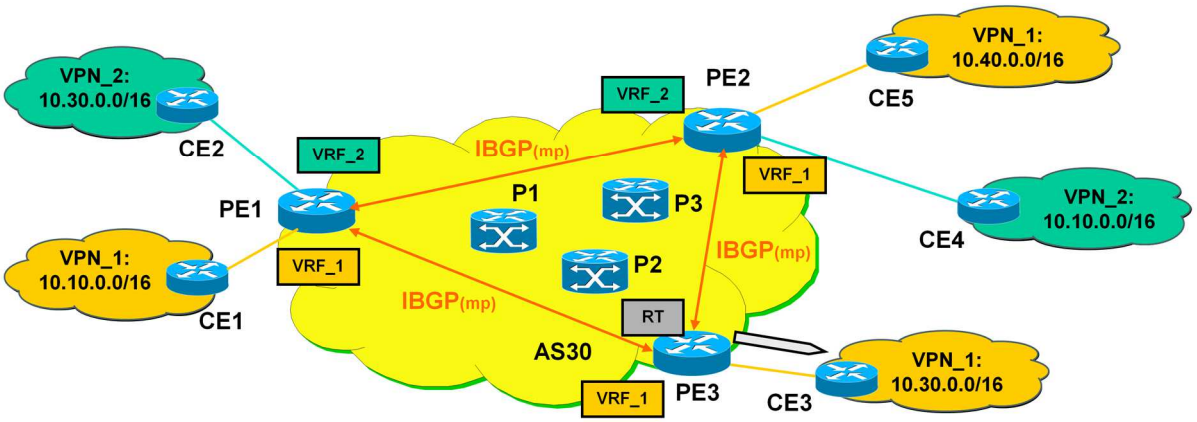
 MPLS-VPN v6.1 56

Appendix 4 - MPLS-VPN (v6.1)



Appendix 4 - MPLS-VPN (v6.1)

New Network 10.10.0.0/16 at PE3/CE3 4



Routing Update from PE1 received at PE 3

VPN-IPv4 update:
RD = 30:1
Net = 10.10/16, Next-Hop = PE1
Label = 3248
RT = Orange

New Route put into VRF_1 based on RT=Orange

VRF_1 (PE3)
10.10/16 via PE1 use 3248

FIB (RT PE3)
PE1 via P2 use 89

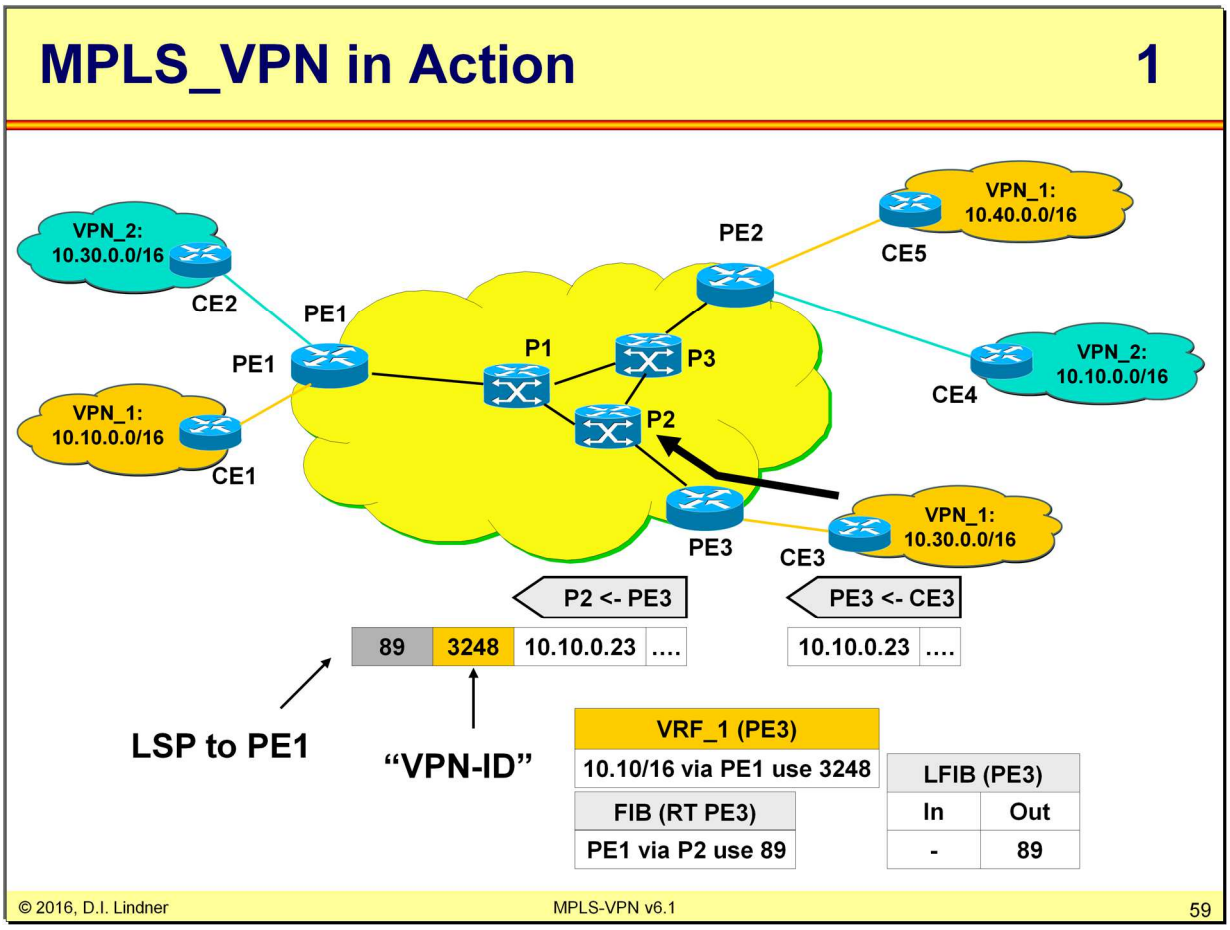
Routing Update to CE3

10.10/16 exist

RT (CE3)
10.10/16 via PE3

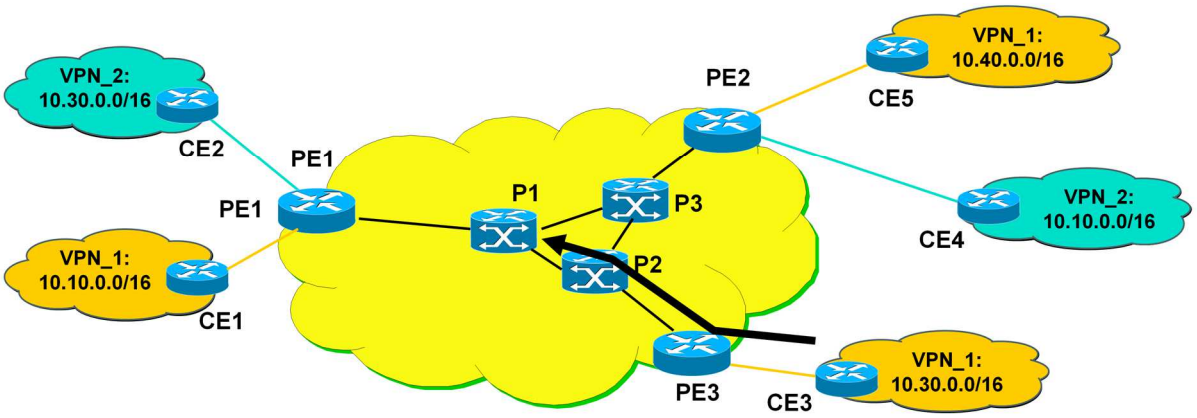
 MPLS-VPN v6.1 58

Appendix 4 - MPLS-VPN (v6.1)



Appendix 4 - MPLS-VPN (v6.1)

MPLS_VPN in Action 2

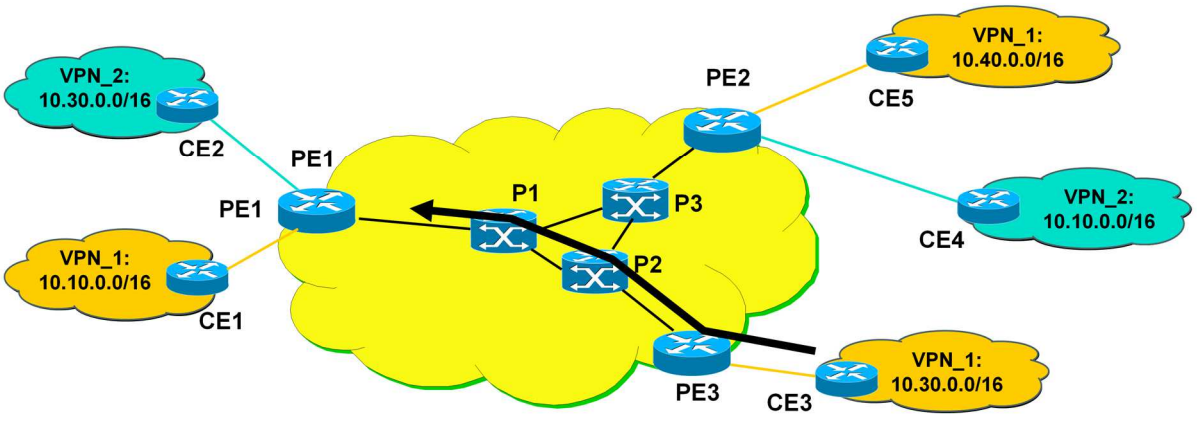


	LFIB (P2)	
FIB (RT P2)	In	Out
PE1 via P1 use 77	89	77

© 2016, D.I. Lindner MPLS-VPN v6.1 60

Appendix 4 - MPLS-VPN (v6.1)

MPLS_VPN in Action 3



PE1 <- P1

3248	10.10.0.23	....

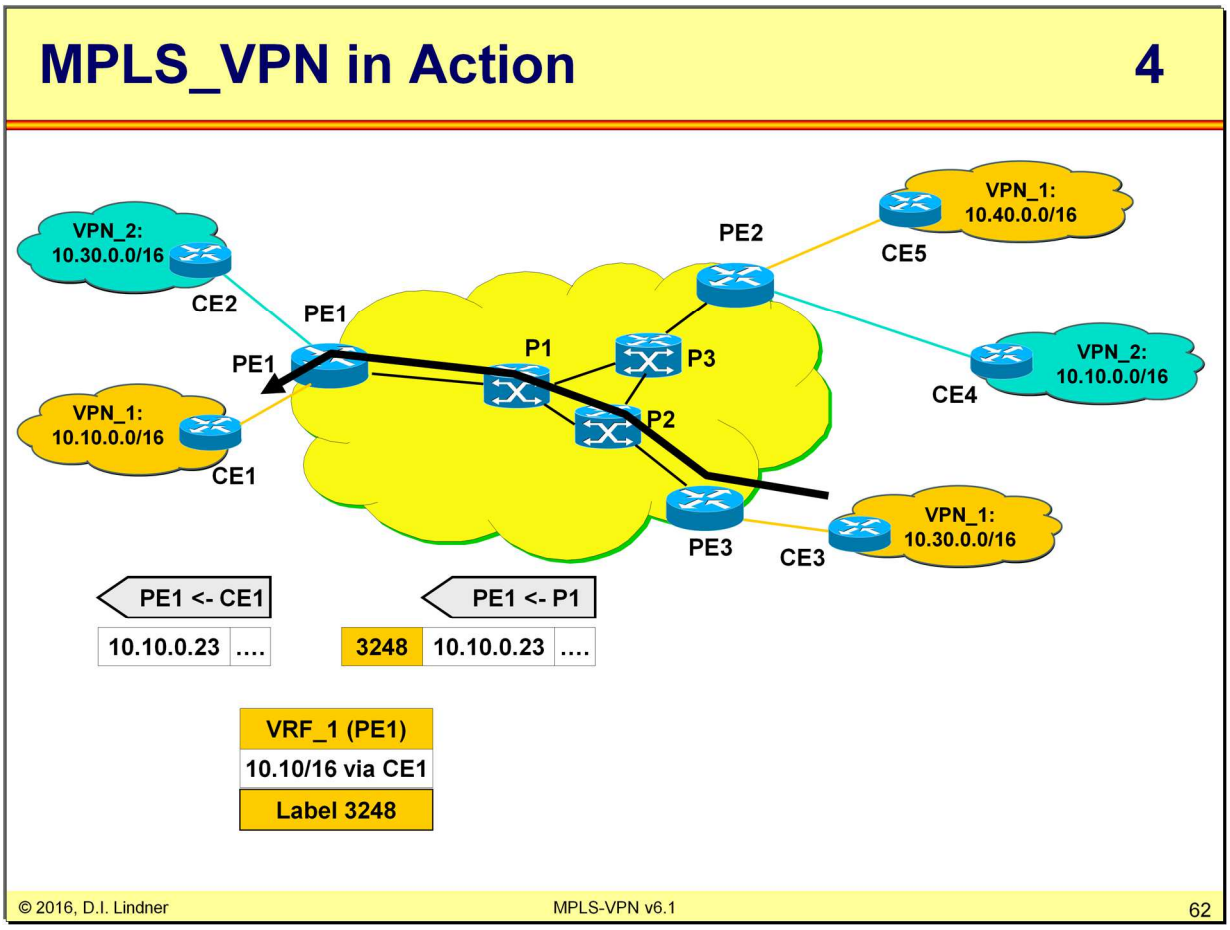
P1 <- P2

77	3248	10.10.0.23	....

FIB (RT P1)	LFIB (P1)	
	In	Out
PE1 via PE1 use null	77	POP

 MPLS-VPN v6.1 61

Appendix 4 - MPLS-VPN (v6.1)



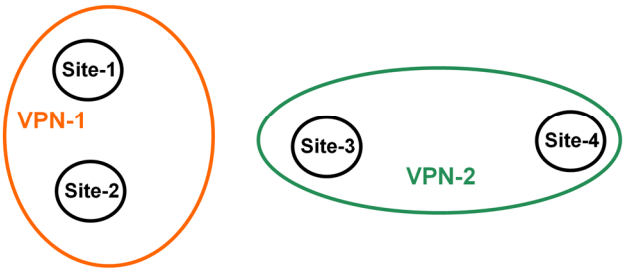
Agenda

- **MP-BGP**
- **VPN Overview**
- **MPLS VPN Architecture**
- **MPLS VPN Basic VPNs**
- **MPLS VPN Complex VPNs**
- **MPLS VPN Configuration (Cisco)**
  - CE-PE OSPF Routing
  - CE-PE Static Routing
  - CE-PE RIP Routing
  - CE-PE External BGP Routing

Appendix 4 - MPLS-VPN (v6.1)

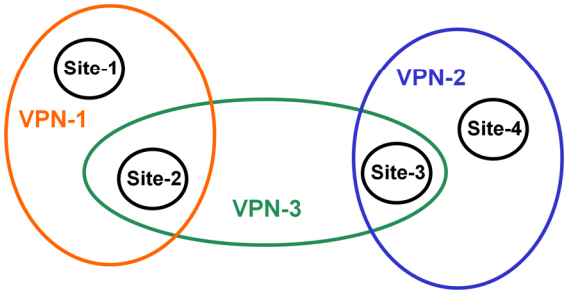
MPLS VPN Types

VPNs not overlapping (Intranet)



site-2 networks can reach
site-1 networks and vice versa,
site-3 networks can reach
site-4 networks and vice versa.

VPNs overlapping (Intranet/Extranet)



site-2 networks can reach
site-1 and site-3 networks,
site-3 networks can reach
site-4 and site-3 networks,
site-1 networks can reach
site-2 networks only,
site-4 networks can reach
site-3 networks only.

 MPLS-VPN v6.1 64

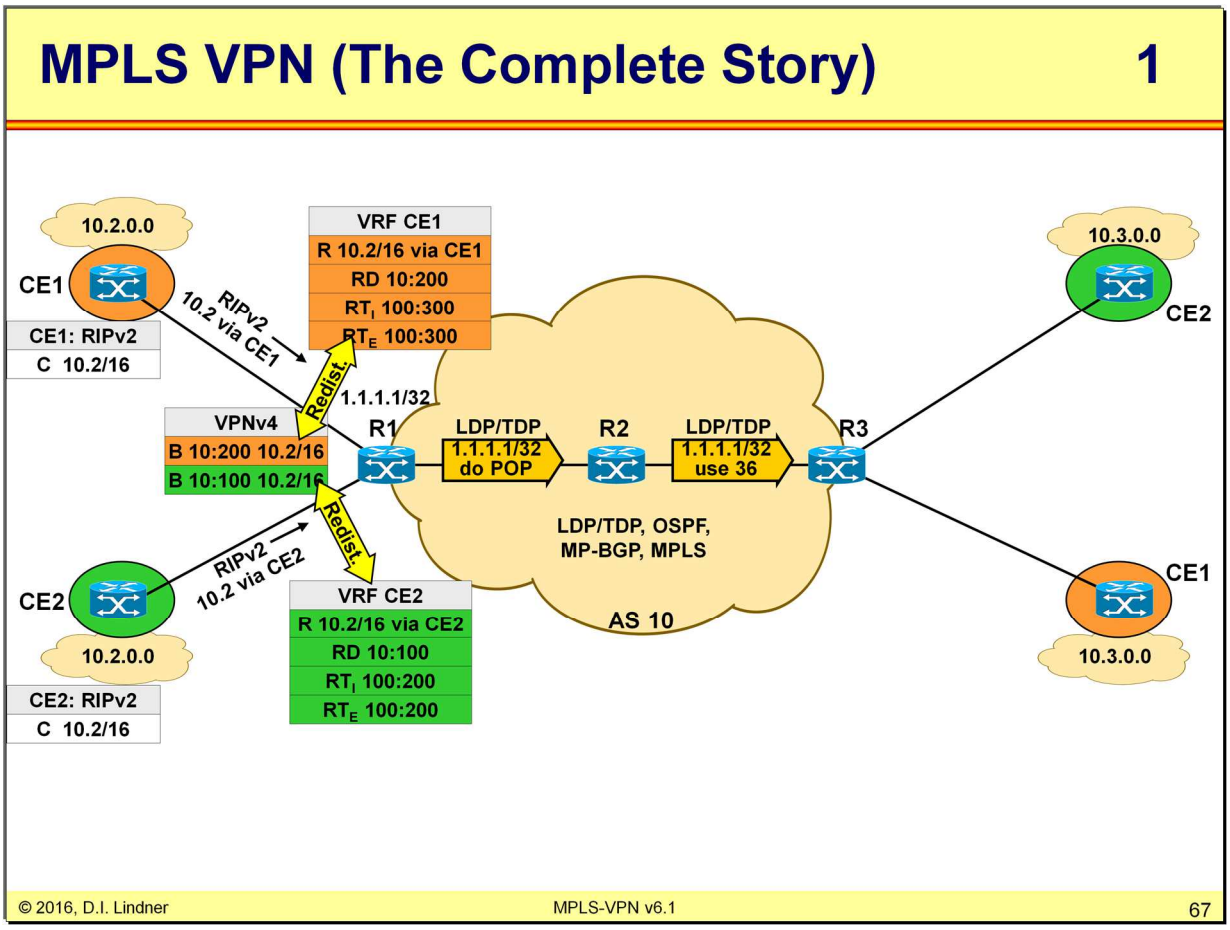
A New Sight of VPN

- **For non-overlapping VPNs**
  - The Route Distinguisher would be sufficient
- **For overlapping VPNs**
  - The Route Distinguisher is not sufficient to achieve the new sight (the Extranet policy) of VPNs
- **In order to implement this new sight of VPNs in case of overlapping VPNs**
  - the Route Target was introduced in the MPLS_VPN Architecture

The real Role of the Route Target

- **PE router which announces a VPNv4 route**
  - uses the Route Target community to specify in which foreign VRFs the announced route should be installed
  - Route Target has export meaning
- **PE router which receives a VPNv4 route**
  - uses the received Route Target community to decide in which local VRFs the announced route should be installed
  - Route Target has import meaning

Appendix 4 - MPLS-VPN (v6.1)



Each interface is exclusively member of the global routing process OR one VRF. The RD, RTi, and RTe are manually configured by the administrator. Each VRF has configured exactly one RD , but can have one or more RTi and RTE. The RD identifies each VPN (unless overlapping VPNs are configured). Routes for a VPNs are learned via an standard routing process running between the PE and the CE router such as RIPv2, OSPF, EIGRP and EBGP.

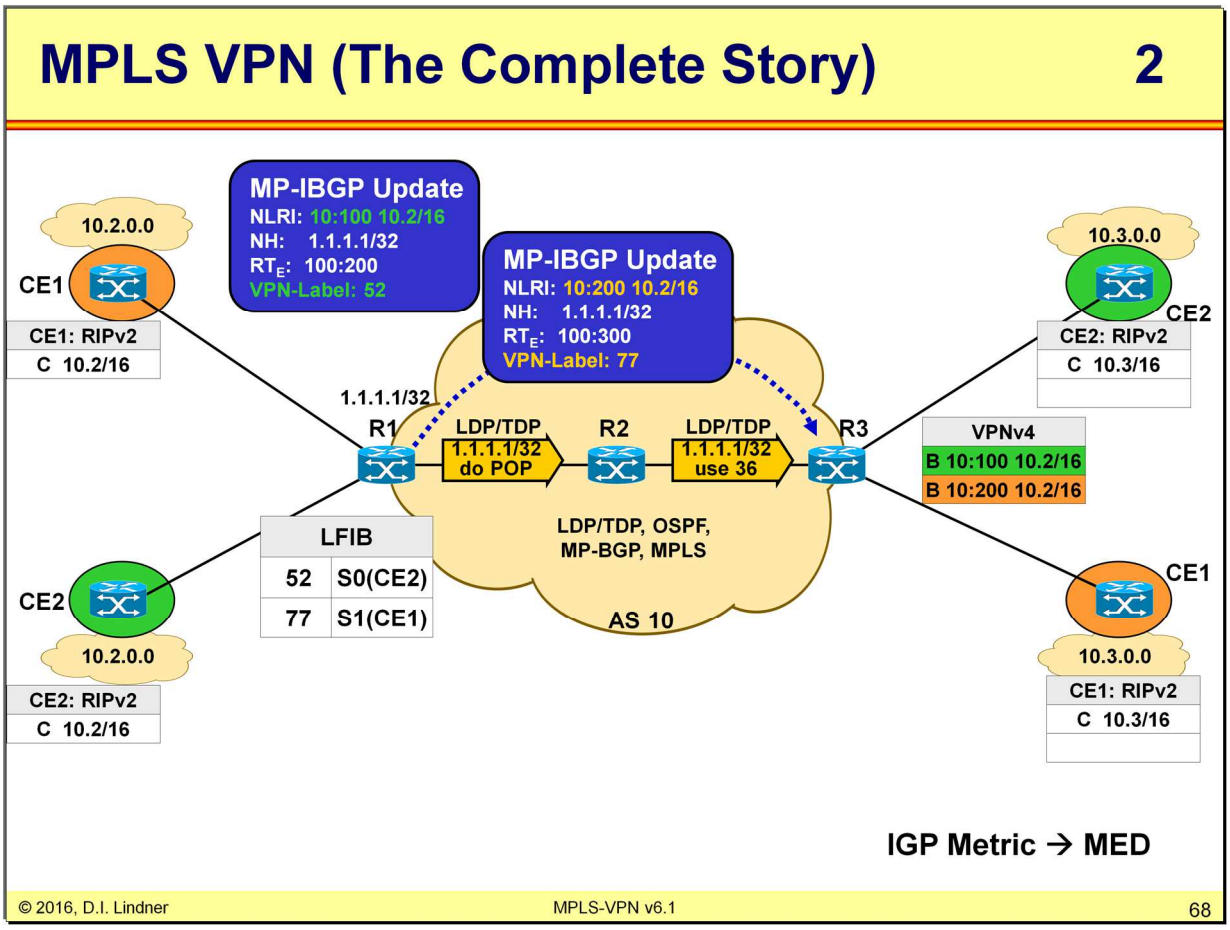
RIPv2, EIGRP or EBGP are good choices because a link state protocol such as OSPF would be limited to approx. 28 processes (theoretically a total of 32 routing processes). RIPv2, EIGRP and EBGP on the other hand can maintain many sub-processes, consuming only one process-number.

Bidirectional redistribution needs to be configured between MP-BGP and OSPF, RIPv2 and EIGRP, which copies the IGP information into the MP-BGP VPNv4 table and vice versa. Redistribution is not needed when EBGP is used as the PE-CE routing protocol.

Learned routes and the preconfigured RD is redistributed from the VRF tables into the MP-BGP VPNv4 table and since BGP makes triggered updates, this information is sent to the peers.

Note: the MP-BGP VPNv4 Table does not show the RTe, but the RTe is copied into to the BGP-database during the redistribution process.

Appendix 4 - MPLS-VPN (v6.1)

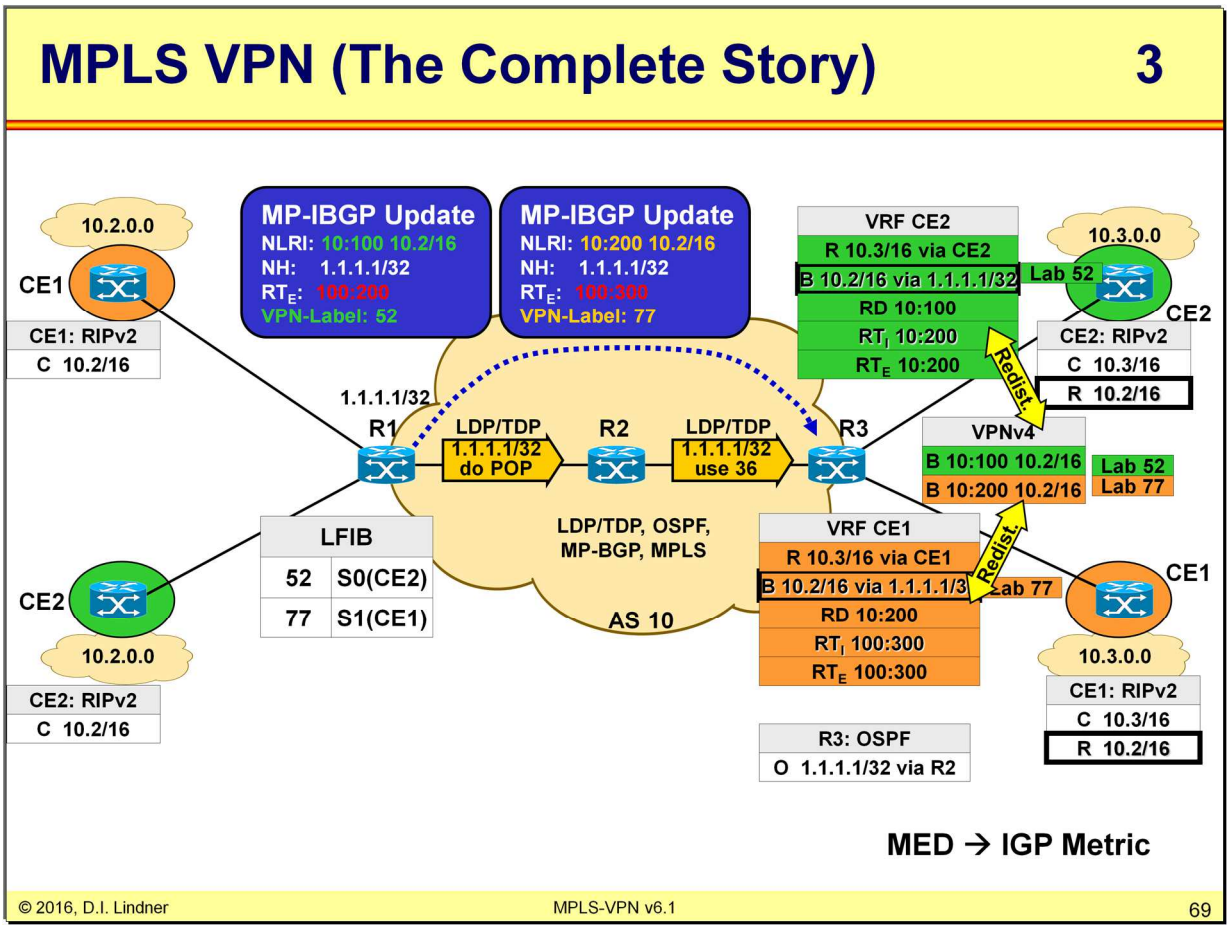


The RD together with the IPv4 address makes up the VPNv4 address which is propagated via MP-BGP updates. These VPNv4 addresses are now used in the NLRI fields of the BGP update instead of traditional IPv4 addresses. Also the RTe is carried with this update using extended community attributes as well as the VPN Label information.

The received MP-IBGP update is then imported into all VRFs which hold a matching RTi and optionally redistributed towards the connected CE routers. During the import from the VPNV4 table to the VRF the RD is removed resulting in a standard IPV4 address.

The IGP Metric (i. e. the RIPv2 hop count) is copied into BGP MED attributes, in order to carry this information to the other side.

Appendix 4 - MPLS-VPN (v6.1)

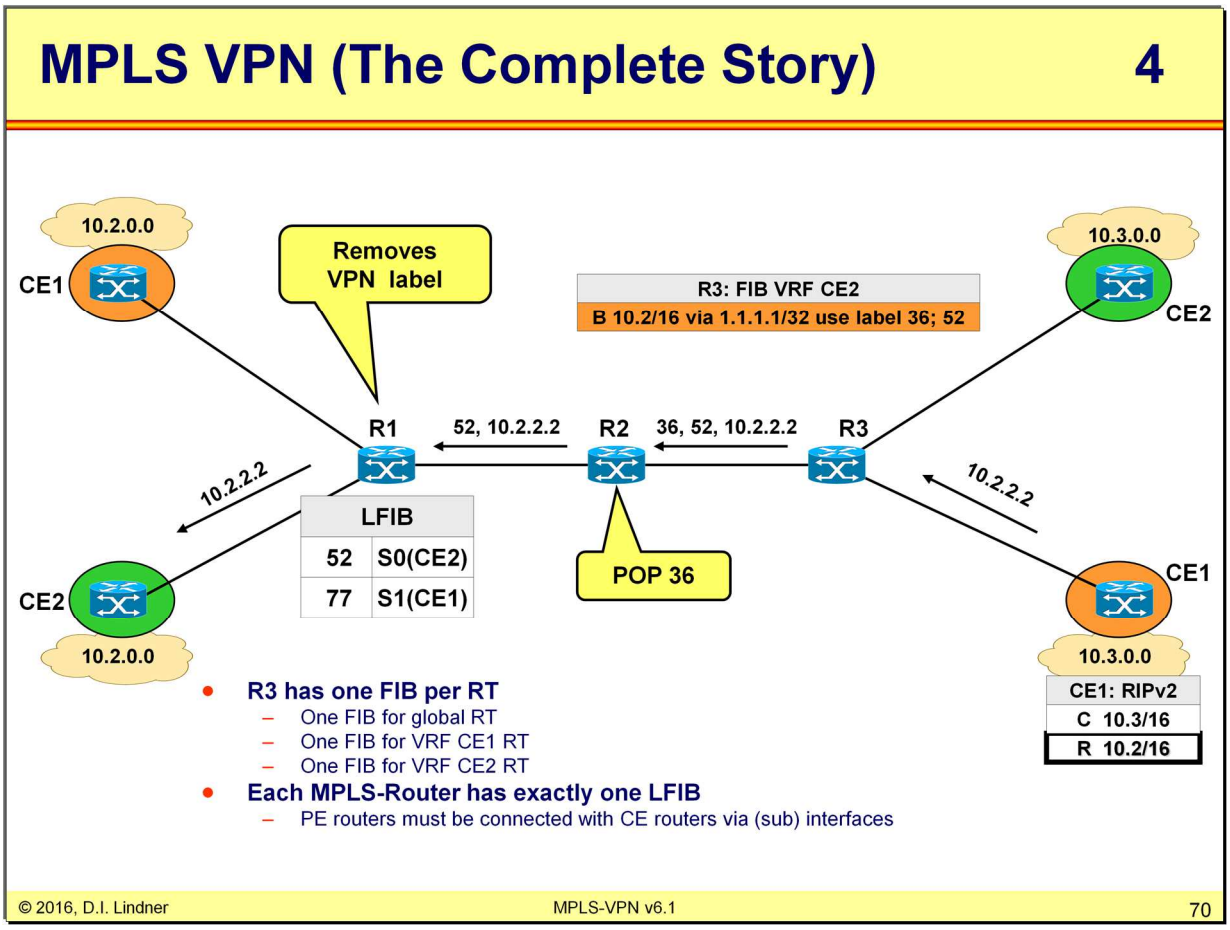


The RTi (import) is used locally by a VRF instance to determine which routes will be imported in the VRF-table and which not.

Routes are only imported into the VRF if the RTe matches the RTi. Also a MPLS-label for this VPN is communicated via IBGP and is directly copied into the CEF table (FIB) of the peer PE router.

The MED attribute is copied into the hop-count field of the RIPv2 update. Thus, CE1 and CE2 on the right side learn about the metric which was specified on the other edge of the provider. The MPLS network is fully transparent to RIPv2 and only increases the IGP metric by one.

Appendix 4 - MPLS-VPN (v6.1)

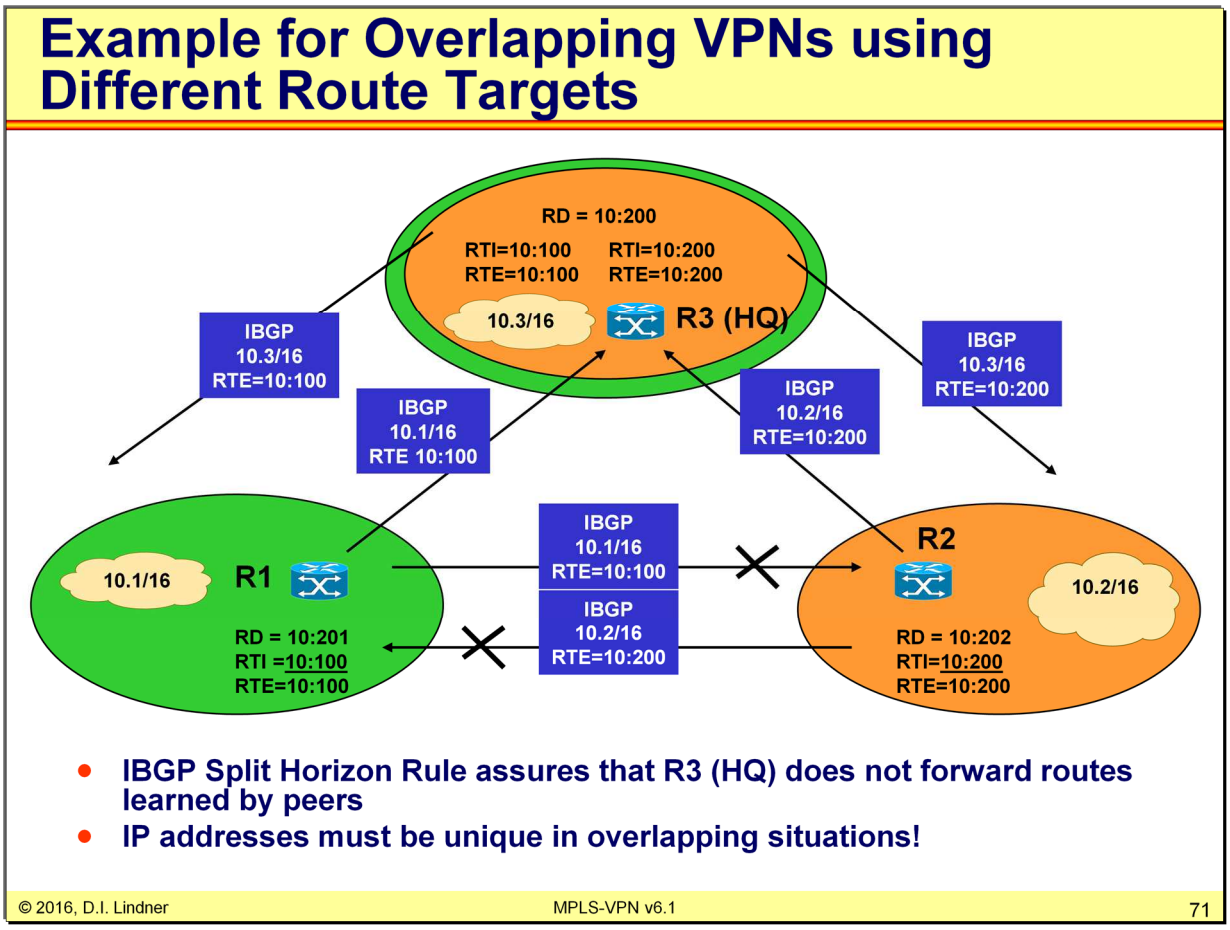


Now IP packets can be forwarded between the VPNs. For example, IP packets to 10.2.2.2 are forwarded from the CE1 router (right side) to the next hop VRF-R3, which adds the labels {36; 52} into the MPLS header, according to its FIB.

R2 pops the MPLS-Transport header and R1 can quickly deliver the IP packet to the correct VPN according to the remaining VPN label {52} which is stored in the LFIB table at R1 pointing to the interface of the appropriate VPN.

R1 removes the MPLS-VPN label {52} before the IP packet is delivered to CE2 (left side). Thus, the VPNs do not recognize any MPLS network in-between; MPLS is completely transparent.

Appendix 4 - MPLS-VPN (v6.1)



When using simple VPNs the RTi is equal to the RTe (keyword "both" when configuring) , but when overlapping VPNs are used, the Route Targets need to be different according to the desired communication behavior.

In our example all routes from the VPN-green and VPN-red are propagated to R3 (HQ) and copied into the VRF table due to the configured RTe and RTi values.

If R3 sends its update towards R1 and R3 all routes (except routes learned from IBGP sessions) out of R3s VRF are propagated to R1 and R2 with both RTEs attached. These routes are then imported by R1 and R2 into the appropriate VRF tables.

Due to the IBGP split horizon rule R3 does not propagate routes learned from R2 towards R3 and vice versa. So without the IBGP split horizon rule MPLS VPNs would not exist.

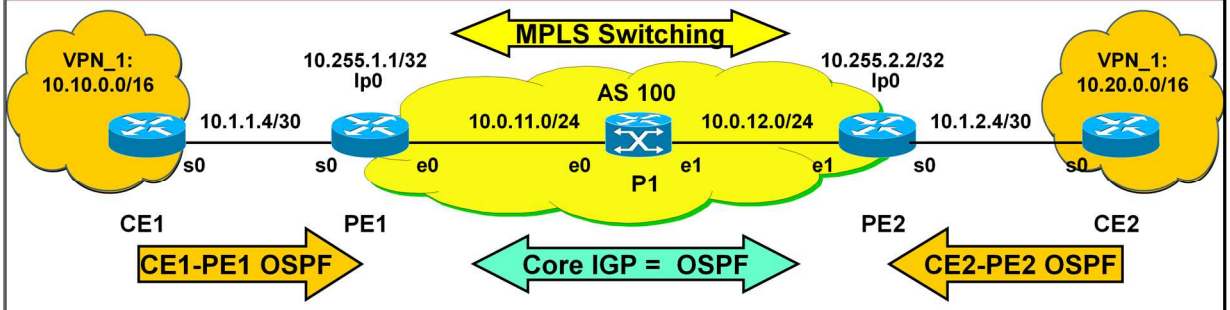
Note: Both RTi and RTe can be configured multiple times. For example one VRF on a router can have specified three different RTi values. Therefore, all IBGP updates whose RTe values match one of the specified RTi values can be imported.

Agenda

- MP-BGP
- VPN Overview
- MPLS VPN Architecture
- MPLS VPN Basic VPNs
- MPLS VPN Complex VPNs
- MPLS VPN Configuration (Cisco)
  - CE-PE OSPF Routing
  - CE-PE Static Routing
  - CE-PE RIP Routing
  - CE-PE External BGP Routing

Appendix 4 - MPLS-VPN (v6.1)

IP Addressing, OSPF Routing in VPN_1, Basic OSPF Routing and MPLS in AS 100



CE1:
```
int s0
  ip address 10.1.1.5 255.255.255.252
router ospf 10
 network 10.1.1.5 0.0.0.0 area 0
```

PE1:
```
ip cef
int loopback  0
  ip address 10.255.1.1 255.255.255.255
int s0
  ip address 10.1.1.6 255.255.255.252
int e0
  ip address 10.0.11.1 255.255.255.0
  mpls ip
router ospf 100
  network 10.0.11.1 0.0.0.0 area 0
```

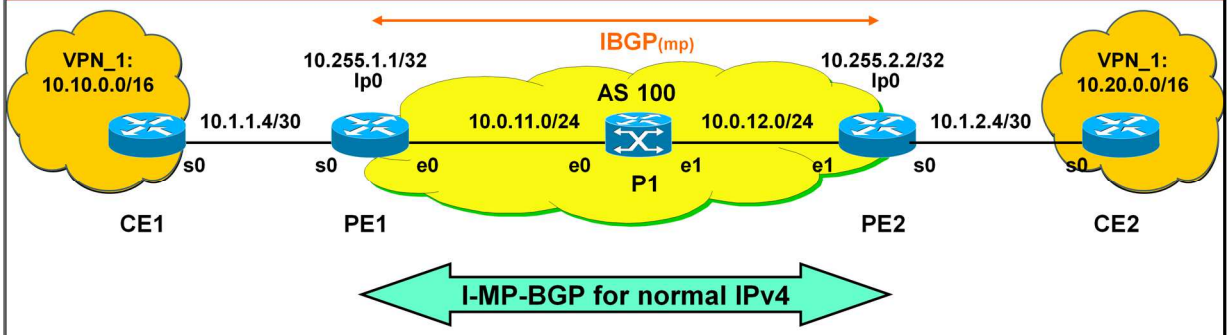
CE2:
```
int s0
  ip address 10.1.2.5 255.255.255.252
router ospf 10
 network 10.1.2.5 0.0.0.0 area 0
```

PE2:
```
ip cef
int loopback  0
  ip address 10.255.2.2 255.255.255.255
int s0
  ip address 10.1.2.6 255.255.255.252
int e0
  ip address 10.0.12.2 255.255.255.0
  mpls ip
router ospf 100
  network 10.0.12.1 0.0.0.0 area 0
```

 MPLS-VPN v6.1 73

Appendix 4 - MPLS-VPN (v6.1)

Start Normal I-BGP in AS 100



I-MP-BGP for normal IPv4

PE1:

```
int loopback  0
  ip address 10.255.1.1 255.255.255.255

router bgp 100
  no bgp default ipv4-unicast
  bgp router-id 10.255.1.1
  neighbor 10.255.2.2 remote-as 100
  neighbor 10.255.2.2 update-source loop 0
  address-family ipv4
    neighbor 10.255.2.2 next-hop-self
    neighbor 10.255.2.2 activate
    no auto-summary (default)
    no synchronization (default)
  exit address-family
```

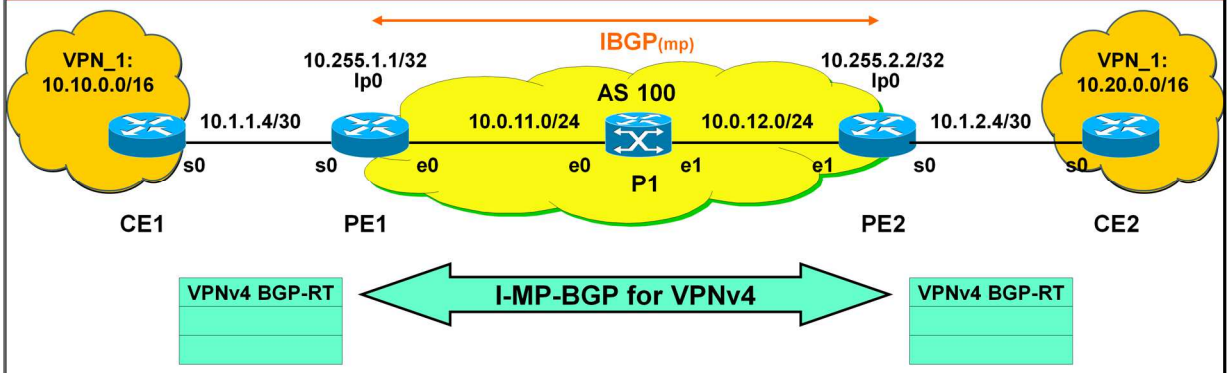
PE2:

```
int loopback  0
  ip address 10.255.2.2 255.255.255.255
router bgp 100
  no bgp default ipv4-unicast
  bgp router-id 10.255.2.2
  neighbor 10.255.1.1 remote-as 100
  neighbor 10.255.1.1 update-source loop 0
  address-family ipv4
    neighbor 10.255.1.1 next-hop-self
    neighbor 10.255.1.1 activate
    no auto-summary (default)
    no synchronization (default)
  exit address-family
```

 MPLS-VPN v6.1 74

Appendix 4 - MPLS-VPN (v6.1)

Start MP-BGP in AS 100



PE1:

```
int loopback  0
  ip address 10.255.1.1 255.255.255.255
router bgp 100
  no bgp default ipv4-unicast
  bgp router-id 10.255.1.1
  neighbor 10.255.2.2 remote-as 100
  neighbor 10.255.2.2 update-source loop 0
 address-family vpnv4
    neighbor 10.255.2.2 activate
    neighbor 10.255.2.2 next-hop-self
    neighbor 10.255.2.2 send-community extended (default)
  exit-address-family
```

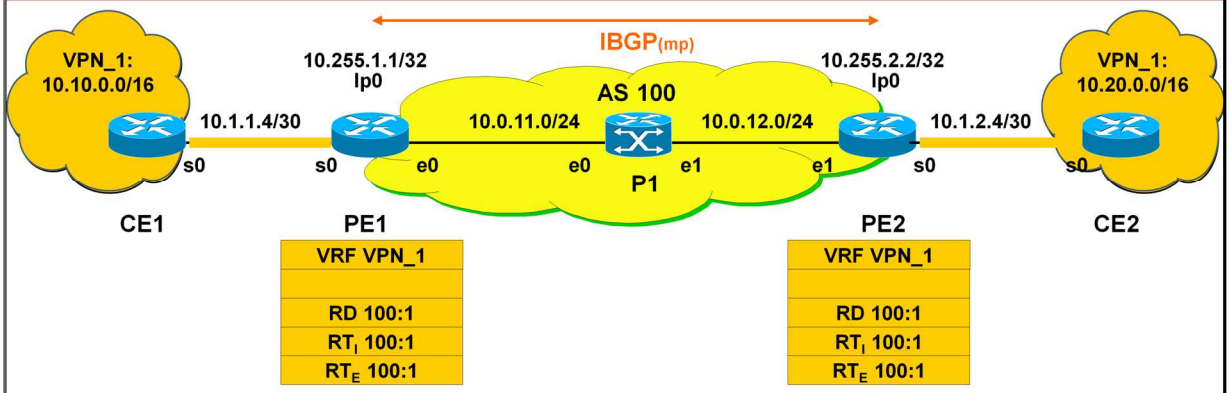
PE2:

```
int loopback  0
  ip address 10.255.2.2 255.255.255.255
router bgp 100
  no bgp default ipv4-unicast
  bgp router-id 10.255.2.2
  neighbor 10.255.1.1 remote-as 100
  neighbor 10.255.1.1 update-source loop 0
  address-family vpnv4
    neighbor 10.255.1.1 activate
    neighbor 10.255.1.1 next-hop-self
    neighbor 10.255.1.1 send-community extended
  exit-address-family
```

 MPLS-VPN v6.1 75

Appendix 4 - MPLS-VPN (v6.1)

Create VRF and Bring Interface into VRF (PE router)



VRF VPN_1
RD 100:1
RT_I 100:1
RT_E 100:1

VRF VPN_1
RD 100:1
RT_I 100:1
RT_E 100:1

PE1:

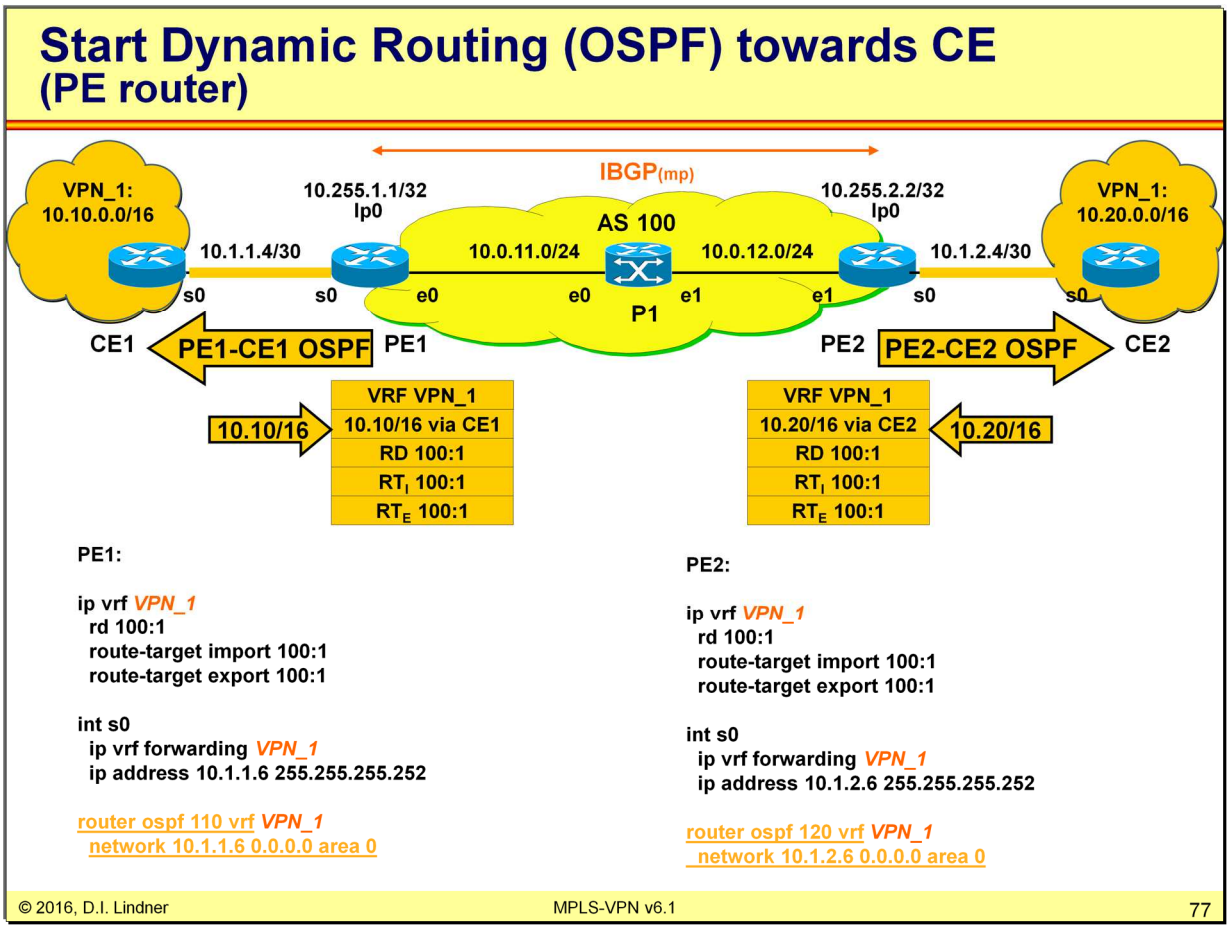
```
ip vrf VPN_1
  rd 100:1
  route-target import 100:1
  route-target export 100:1

int s0
  ip vrf forwarding VPN_1
  ip address 10.1.1.6 255.255.255.252
```

PE2:

```
ip vrf VPN_1
  rd 100:1
  route-target import 100:1
  route-target export 100:1

int s0
  ip vrf forwarding VPN_1
  ip address 10.1.2.6 255.255.255.252
```

 MPLS-VPN v6.1 76

Appendix 4 - MPLS-VPN (v6.1)



!!! Router ospf xxx-number vrf VPN_1 !!!

same OSPF process number on both sides (PE1, PE2):

MPLS -> VPN Superbackbone area0 -> networks of other site are imported as OSPF internal IA routes, PE becomes ABR

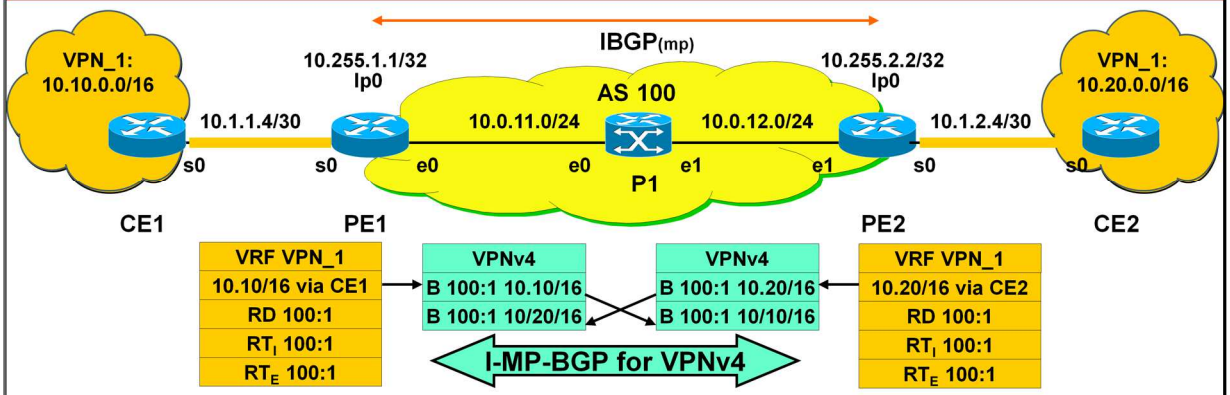
different OSPF process number on every side (PE1, PE2):

MPLS -> VPN Superbackbone area 0 -> networks of other site are imported as OSPF external E2 routes, PE becomes ASBR

Information about OSPF is transported by a new extendend community attribute: -> OPSF-ID, RT LSA-Type (2 = 0, 3 = IA)

Appendix 4 - MPLS-VPN (v6.1)

Redistributing VRF OSPF into MP-BGP and Transport of VPNv4 routes via I-MP-BGP (PE router)



VRF VPN_1	VPNv4	VPNv4	VRF VPN_1
10.10/16 via CE1	B 100:1 10.10/16	B 100:1 10.20/16	10.20/16 via CE2
RD 100:1	B 100:1 10/20/16	B 100:1 10/10/16	RD 100:1
RT_I 100:1			RT_I 100:1
RT_E 100:1			RT_E 100:1

I-MP-BGP for VPNv4

PE1:

```
ip vrf VPN_1
 rd 100:1

router ospf 110 vrf VPN_1
 network 10.1.1.6 0.0.0.0 area 0

router bgp 100
  no bgp default ipv4-unicast
  address-family ipv4 vrf VPN_1
   redistribute ospf 110 match internal
    no auto-summary
    no synchronization
  exit-address-family
```

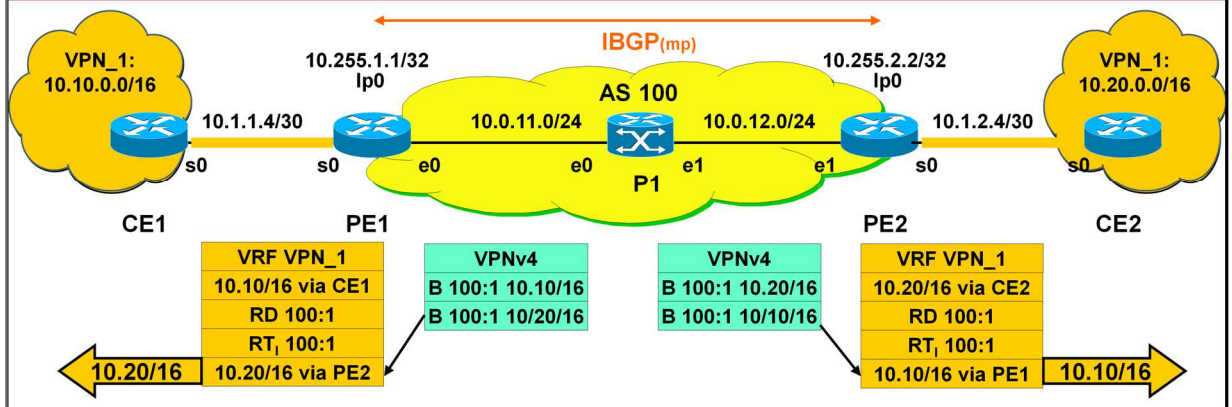
PE2:

```
ip vrf VPN_1
 rd 100:1

router ospf 120 vrf VPN_1
 network 10.1.2.6 0.0.0.0 area 0

router bgp 100
  no bgp default ipv4-unicast
  address-family ipv4 vrf VPN_1
   redistribute ospf 120 match internal
    no auto-summary
    no synchronization
  exit-address-family
```

 MPLS-VPN v6.1 78

Appendix 4 - MPLS-VPN (v6.1)

Redistribution of VPNv4 routes into VRF OSPF (PE router)



PE1:

```
router ospf 110 vrf VPN_1
  network 10.1.1.6 0.0.0.0 area 0
  redistribute bgp 100 metric <20> subnet

router bgp 100
  no bgp default ipv4-unicast
  address-family ipv4 vrf VPN_1
    redistribute ospf 100 match internal
    no auto-summary
  exit-address-family
```

PE2:

```
router ospf 120 vrf VPN_1
  network 10.1.2.6 0.0.0.0 area 0
  redistribute bgp 100 metric <20> subnets

router bgp 100
  no bgp default ipv4-unicast
  address-family ipv4 vrf VPN_1
    redistribute ospf 100 match internal
    no auto-summary
  exit-address-family
```

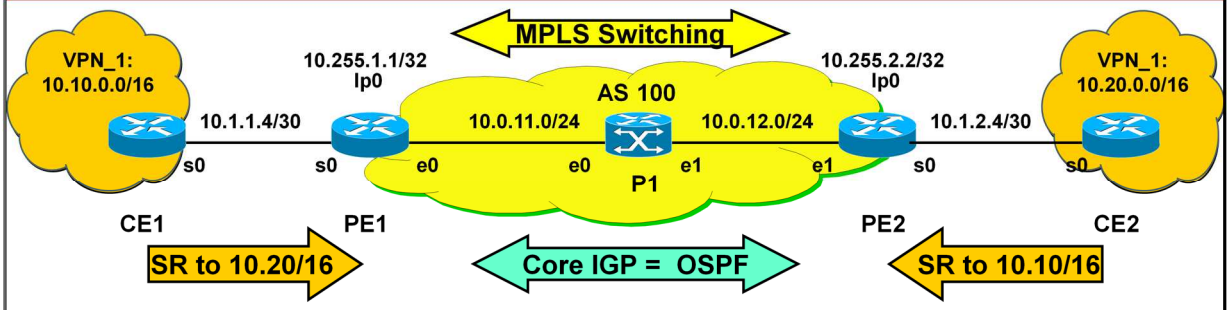
 MPLS-VPN v6.1 79

Agenda

- **MP-BGP**
- **VPN Overview**
- **MPLS VPN Architecture**
- **MPLS VPN Basic VPNs**
- **MPLS VPN Complex VPNs**
- **MPLS VPN Configuration (Cisco)**
  - CE-PE OSPF Routing
  - CE-PE Static Routing
  - CE-PE RIP Routing
  - CE-PE External BGP Routing

Appendix 4 - MPLS-VPN (v6.1)

IP Addressing, Static Routing in VPN_1, Basic OSPF Routing and MPLS in AS 100



CE1:

```
int s0
  ip address 10.1.1.5 255.255.255.252
  ip route 10.20.0.0 255.255.0.0 10.1.1.6
```

PE1 (OSPF and MPLS in Backbone):

```
ip cef
int loopback  0
  ip address 10.255.1.1 255.255.255.255
int s0
  ip address 10.1.1.6 255.255.255.252
int e0
  ip address 10.0.11.1 255.255.255.0
  mpls ip
router ospf 100
  network 10.0.11.1 0.0.0.0 area 0
```

CE2:

```
int s0
  ip address 10.1.2.5 255.255.255.252
  ip route 10.10.0.0 255.255.0.0 10.1.2.6
```

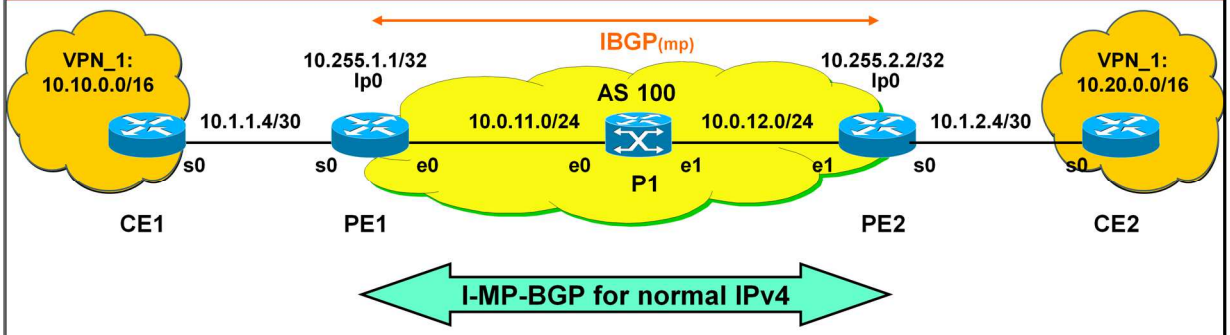
PE2 (OSPF and MPLS in Backbone):

```
ip cef
int loopback  0
  ip address 10.255.2.2 255.255.255.255
int s0
  ip address 10.1.2.6 255.255.255.252
int e0
  ip address 10.0.12.2 255.255.255.0
  mpls ip
router ospf 100
  network 10.0.12.1 0.0.0.0 area 0
```

 MPLS-VPN v6.1 81

Appendix 4 - MPLS-VPN (v6.1)

Start Normal I-BGP in AS 100



I-MP-BGP for normal IPv4

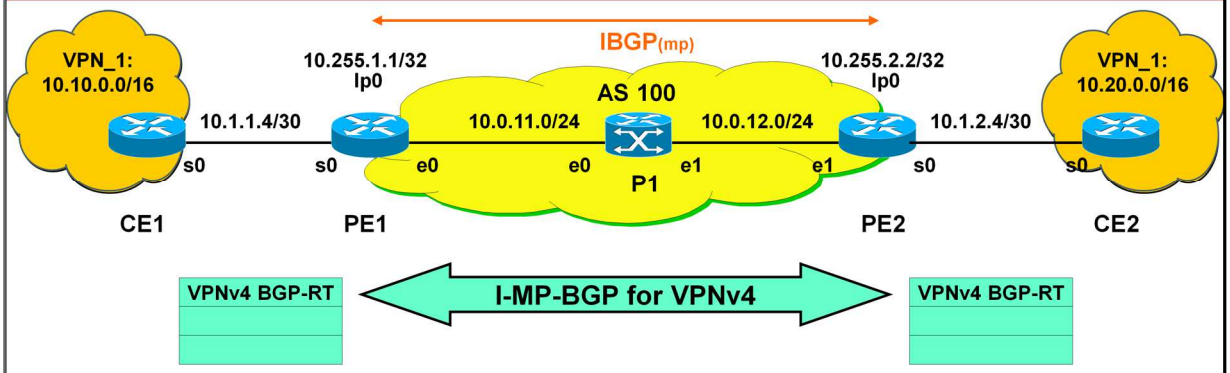
```
PE1:
int loopback  0
  ip address 10.255.1.1 255.255.255.255

router bgp 100
  no bgp default ipv4-unicast
  bgp router-id 10.255.1.1
  neighbor 10.255.2.2 remote-as 100
  neighbor 10.255.2.2 update-source loop 0
  address-family ipv4
   neighbor 10.255.2.2 next-hop-self
   neighbor 10.255.2.2 activate
   no auto-summary (default)
   no synchronization (default)
  exit address-family
```

```
PE2:
int loopback  0
  ip address 10.255.2.2 255.255.255.255
router bgp 100
  no bgp default ipv4-unicast
  bgp router-id 10.255.2.2
  neighbor 10.255.1.1 remote-as 100
  neighbor 10.255.1.1 update-source loop 0
  address-family ipv4
   neighbor 10.255.1.1 next-hop-self
   neighbor 10.255.1.1 activate
   no auto-summary (default)
   no synchronization (default)
  exit address-family
```

 MPLS-VPN v6.1 82

Appendix 4 - MPLS-VPN (v6.1)

Start MP-BGP in AS 100



PE1:

```
int loopback 0
  ip address 10.255.1.1 255.255.255.255
router bgp 100
  no bgp default ipv4-unicast
  bgp router-id 10.255.1.1
  neighbor 10.255.2.2 remote-as 100
  neighbor 10.255.2.2 update-source loop 0
 address-family vpnv4
   neighbor 10.255.2.2 activate
   neighbor 10.255.2.2 next-hop-self
   neighbor 10.255.2.2 send-community extended (default)
 exit-address-family
```

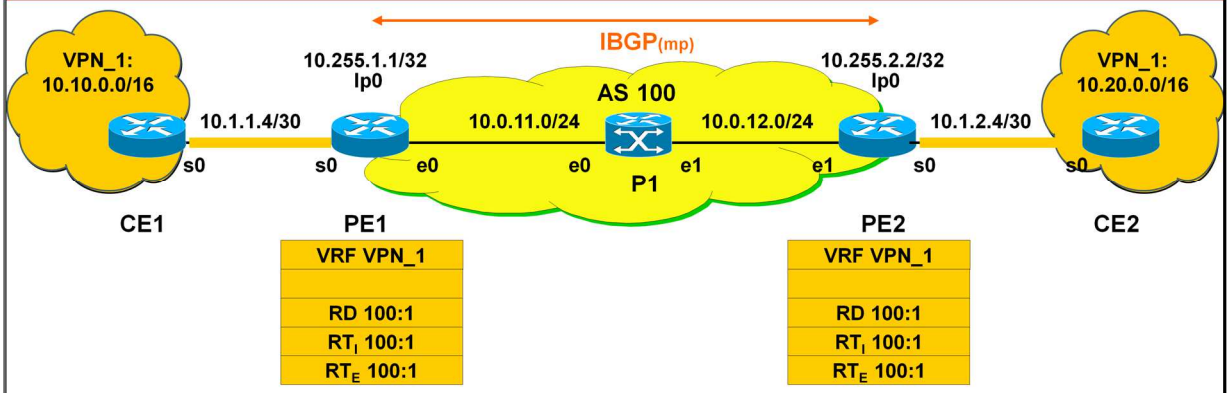
PE2:

```
int loopback 0
  ip address 10.255.2.2 255.255.255.255
router bgp 100
  no bgp default ipv4-unicast
  bgp router-id 10.255.2.2
  neighbor 10.255.1.1 remote-as 100
  neighbor 10.255.1.1 update-source loop 0
  address-family vpnv4
    neighbor 10.255.1.1 activate
    neighbor 10.255.1.1 next-hop-self
    neighbor 10.255.1.1 send-community extended
  exit-address-family
```

 MPLS-VPN v6.1 83

Appendix 4 - MPLS-VPN (v6.1)

Create VRF and Bring Interface into VRF (PE router)



PE1:

```
ip vrf VPN_1
  rd 100:1
  route-target import 100:1
  route-target export 100:1

int s0
  ip vrf forwarding VPN_1
  ip address 10.1.1.6 255.255.255.252
```

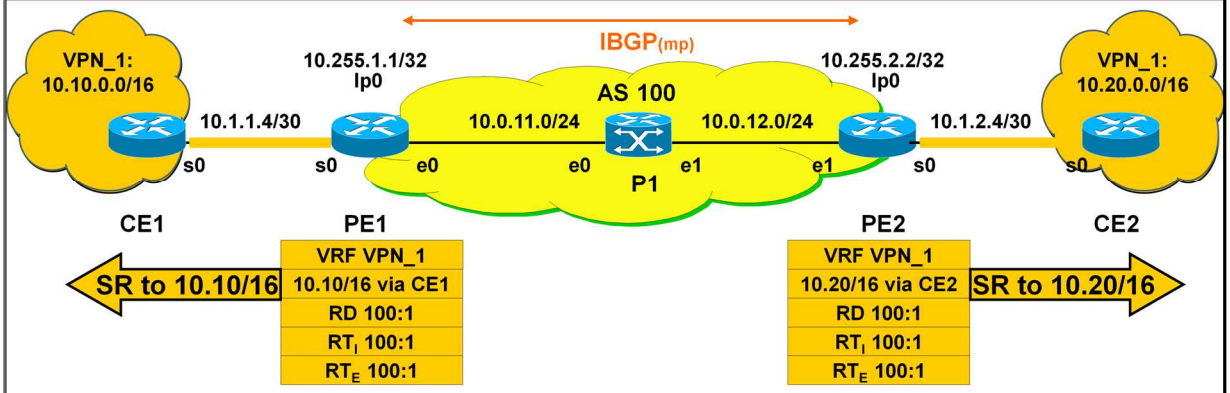
PE2:

```
ip vrf VPN_1
  rd 100:1
  route-target import 100:1
  route-target export 100:1

int s0
  ip vrf forwarding VPN_1
  ip address 10.1.2.6 255.255.255.252
```

 MPLS-VPN v6.1 84

Appendix 4 - MPLS-VPN (v6.1)

Static Routing (SR) towards CE (PE router)



PE1:

```
ip vrf VPN_1
 rd 100:1
 route-target import 100:1
 route-target export 100:1

int s0
 ip vrf forwarding VPN_1
 ip address 10.1.1.6 255.255.255.252

ip route vrf VPN_1  10.10.0.0 255.255.0.0 serial 0 10.1.1.5
```

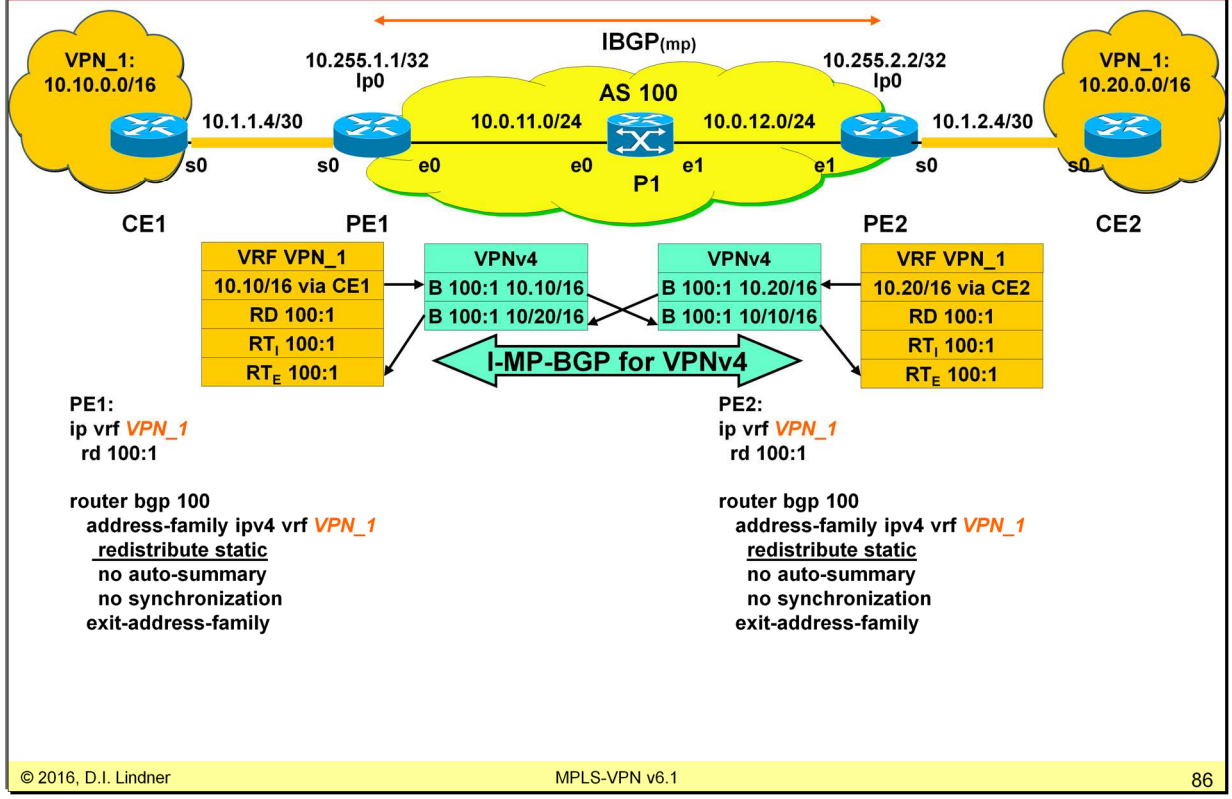
PE2:

```
ip vrf VPN_1
 rd 100:1
 route-target import 100:1
 route-target export 100:1

int s0
 ip vrf forwarding VPN_1
 ip address 10.1.2.6 255.255.255.252

ip route vrf VPN_1  10.20.0.0 255.255.0.0 serial 0 10.1.2.5
```

 MPLS-VPN v6.1 85

Appendix 4 - MPLS-VPN (v6.1)

Redistributing Static into MP-BGP and Transport of Static routes via I-MP-BGP (PE router)



PE1:

```
ip vrf VPN_1
 rd 100:1

router bgp 100
  address-family ipv4 vrf VPN_1
   redistribute static
    no auto-summary
    no synchronization
  exit-address-family
```

PE2:

```
ip vrf VPN_1
 rd 100:1

router bgp 100
  address-family ipv4 vrf VPN_1
   redistribute static
    no auto-summary
    no synchronization
  exit-address-family
```

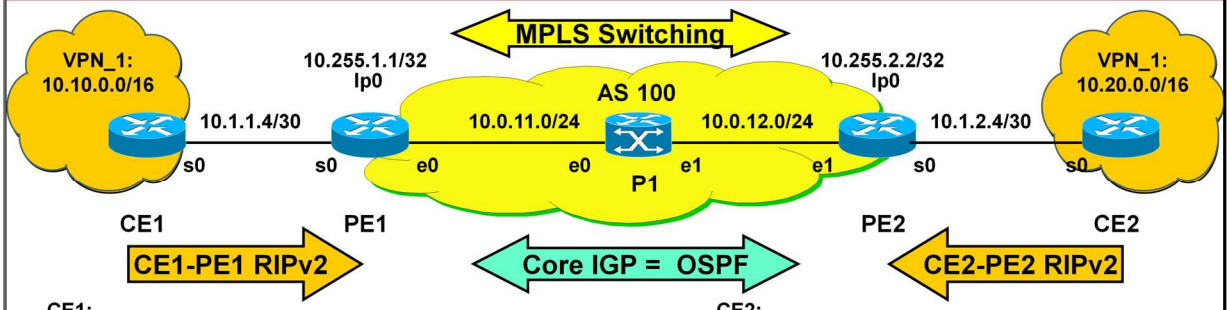
 MPLS-VPN v6.1 86

Agenda

- **MP-BGP**
- **VPN Overview**
- **MPLS VPN Architecture**
- **MPLS VPN Basic VPNs**
- **MPLS VPN Complex VPNs**
- **MPLS VPN Configuration (Cisco)**
  - CE-PE OSPF Routing
  - CE-PE Static Routing
  - CE-PE RIP Routing
  - CE-PE External BGP Routing

Appendix 4 - MPLS-VPN (v6.1)

IP Addressing, RIPv2 Routing in VPN_1, Basic OSPF Routing and MPLS in AS 100



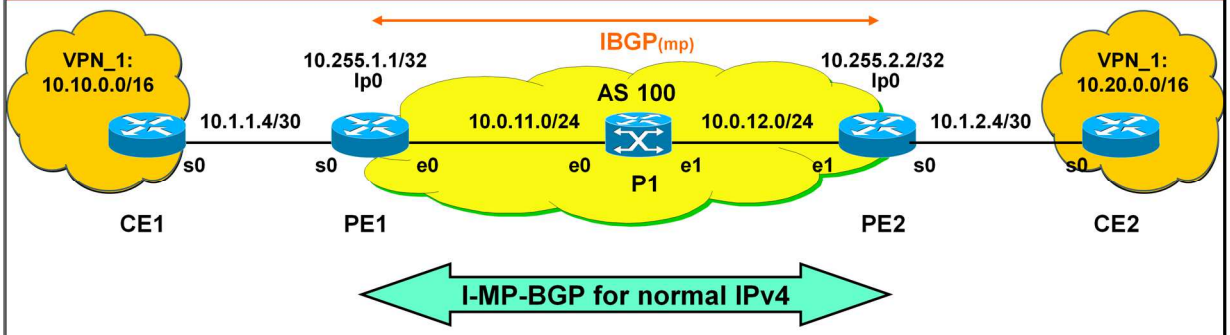

```
CE1:
int s0
  ip address 10.1.1.5 255.255.255.252
router rip
  version 2
  network 10.0.0.0
  no auto-summary

PE1:
ip cef
int loopback  0
  ip address 10.255.1.1 255.255.255.255
int s0
  ip address 10.1.1.6 255.255.255.252
int e0
  ip address 10.0.11.1 255.255.255.0
  mpls ip
router ospf 100
  network 10.0.11.1 0.0.0.0 area 0
```

```
CE2:
int s0
  ip address 10.1.2.5 255.255.255.252
router rip
  version 2
  network 10.0.0.0
  no auto-summary

PE2:
ip cef
int loopback  0
  ip address 10.255.2.2 255.255.255.255
int s0
  ip address 10.1.2.6 255.255.255.252
int e0
  ip address 10.0.12.2 255.255.255.0
  mpls ip
router ospf 100
  network 10.0.12.1 0.0.0.0 area 0
```

 MPLS-VPN v6.1 88

Appendix 4 - MPLS-VPN (v6.1)

Start Normal I-BGP in AS 100



I-MP-BGP for normal IPv4

PE1:

```
int loopback  0
  ip address 10.255.1.1 255.255.255.255

router bgp 100
  no bgp default ipv4-unicast
  bgp router-id 10.255.1.1
  neighbor 10.255.2.2 remote-as 100
  neighbor 10.255.2.2 update-source loop 0
  address-family ipv4
    neighbor 10.255.2.2 next-hop-self
    neighbor 10.255.2.2 activate
    no auto-summary (default)
    no synchronization (default)
  exit address-family
```

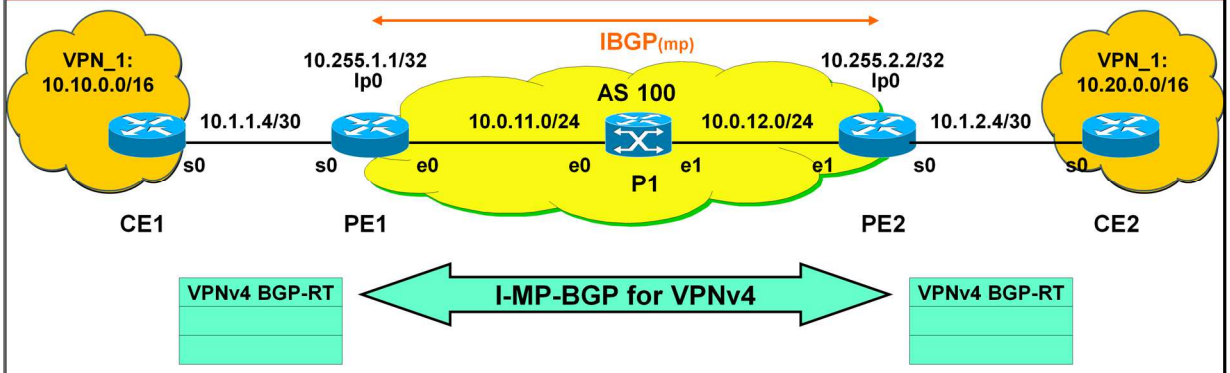
PE2:

```
int loopback  0
  ip address 10.255.2.2 255.255.255.255
router bgp 100
  no bgp default ipv4-unicast
  bgp router-id 10.255.2.2
  neighbor 10.255.1.1 remote-as 100
  neighbor 10.255.1.1 update-source loop 0
  address-family ipv4
    neighbor 10.255.1.1 next-hop-self
    neighbor 10.255.1.1 activate
    no auto-summary (default)
    no synchronization (default)
  exit address-family
```

 MPLS-VPN v6.1 89

Appendix 4 - MPLS-VPN (v6.1)

Start MP-BGP in AS 100



PE1:

```
int loopback  0
  ip address 10.255.1.1 255.255.255.255
router bgp 100
  no bgp default ipv4-unicast
  bgp router-id 10.255.1.1
  neighbor 10.255.2.2 remote-as 100
  neighbor 10.255.2.2 update-source loop 0
 address-family vpnv4
    neighbor 10.255.2.2 activate
    neighbor 10.255.2.2 next-hop-self
    neighbor 10.255.2.2 send-community extended (default)
  exit-address-family
```

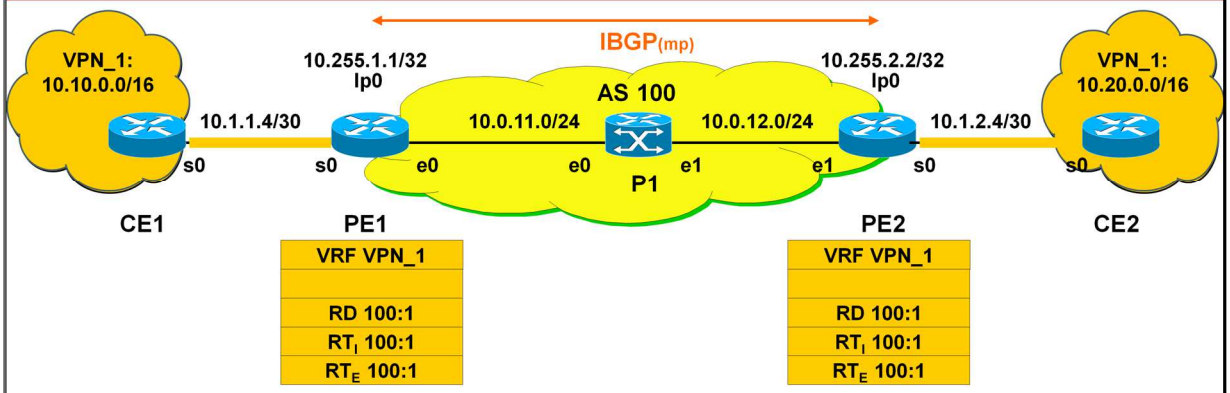
PE2:

```
int loopback  0
  ip address 10.255.2.2 255.255.255.255
router bgp 100
  no bgp default ipv4-unicast
  bgp router-id 10.255.2.2
  neighbor 10.255.1.1 remote-as 100
  neighbor 10.255.1.1 update-source loop 0
  address-family vpnv4
    neighbor 10.255.1.1 activate
    neighbor 10.255.1.1 next-hop-self
    neighbor 10.255.1.1 send-community extended
  exit-address-family
```

 MPLS-VPN v6.1 90

Appendix 4 - MPLS-VPN (v6.1)

Create VRF and Bring Interface into VRF (PE router)



VRF VPN_1
RD 100:1
RT_I 100:1
RT_E 100:1

VRF VPN_1
RD 100:1
RT_I 100:1
RT_E 100:1

PE1:

```
ip vrf VPN_1
  rd 100:1
  route-target import 100:1
  route-target export 100:1

int s0
  ip vrf forwarding VPN_1
  ip address 10.1.1.6 255.255.255.252
```

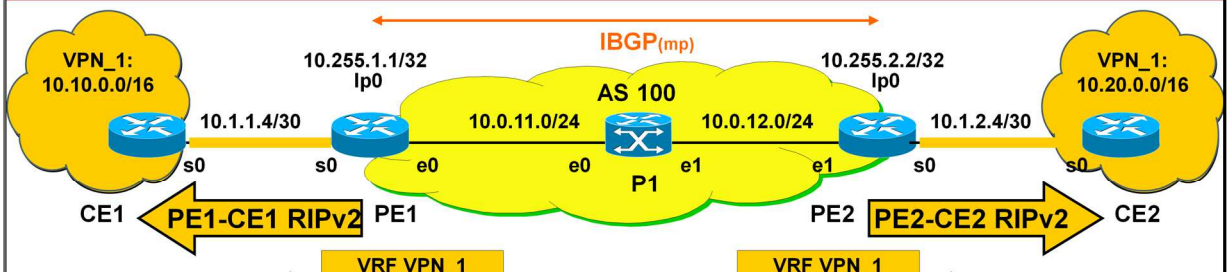
PE2:

```
ip vrf VPN_1
  rd 100:1
  route-target import 100:1
  route-target export 100:1

int s0
  ip vrf forwarding VPN_1
  ip address 10.1.2.6 255.255.255.252
```

 MPLS-VPN v6.1 91

Appendix 4 - MPLS-VPN (v6.1)

Start Dynamic Routing (RIPv2) towards CE (PE router)



VRF VPN_1
10.10/16 via CE1
RD 100:1
RT_I 100:1
RT_E 100:1

10.10/16

VRF VPN_1
10.20/16 via CE2
RD 100:1
RT_I 100:1
RT_E 100:1

10.20/16

PE1:

```
ip vrf VPN_1
 rd 100:1

int s0
 ip vrf forwarding VPN_1
 ip address 10.1.1.6 255.255.255.252

router rip
 version 2
 address-family ipv4 vrf VPN_1
   network 10.0.0.0
   no auto-summary
 exit-address-family
```

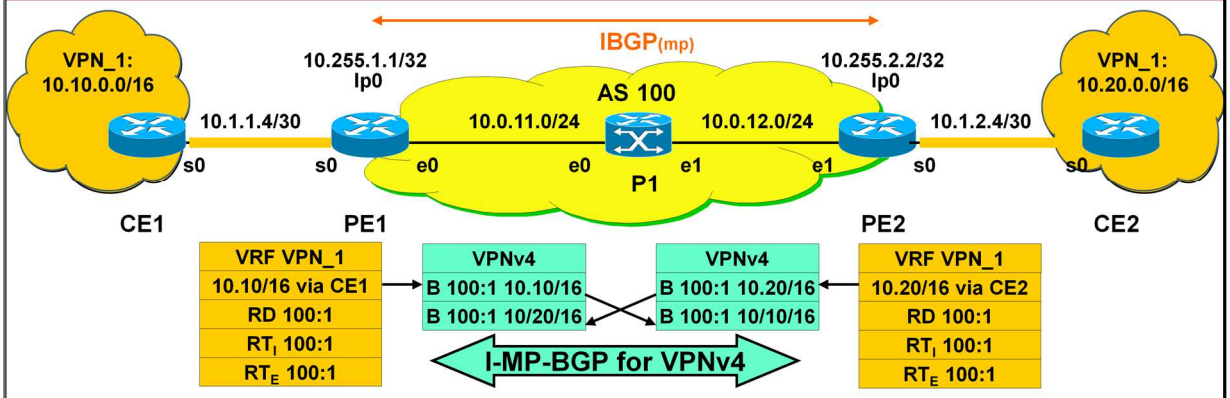
PE2:

```
ip vrf VPN_1
 rd 100:1

int s0
 ip vrf forwarding VPN_1
 ip address 10.1.2.6 255.255.255.252

router rip
 version 2
 address-family ipv4 vrf VPN_1
   network 10.0.0.0
   no auto-summary
 exit-address-family
```

 MPLS-VPN v6.1 92

Appendix 4 - MPLS-VPN (v6.1)

Redistributing VRF RIPv2 into MP-BGP and Transport of VPNv4 routes via I-MP-BGP (PE router)



VRF VPN_1	VPNv4	VPNv4	VRF VPN_1
10.10/16 via CE1	B 100:1 10.10/16	B 100:1 10.20/16	10.20/16 via CE2
RD 100:1	B 100:1 10/20/16	B 100:1 10/10/16	RD 100:1
RT_I 100:1			RT_I 100:1
RT_E 100:1			RT_E 100:1

I-MP-BGP for VPNv4

PE1:
ip vrf *VPN_1*
rd 100:1

router bgp 100
no bgp default ipv4-unicast
address-family ipv4 vrf *VPN_1*
redistribute rip
no auto-summary
no synchronization
exit-address-family

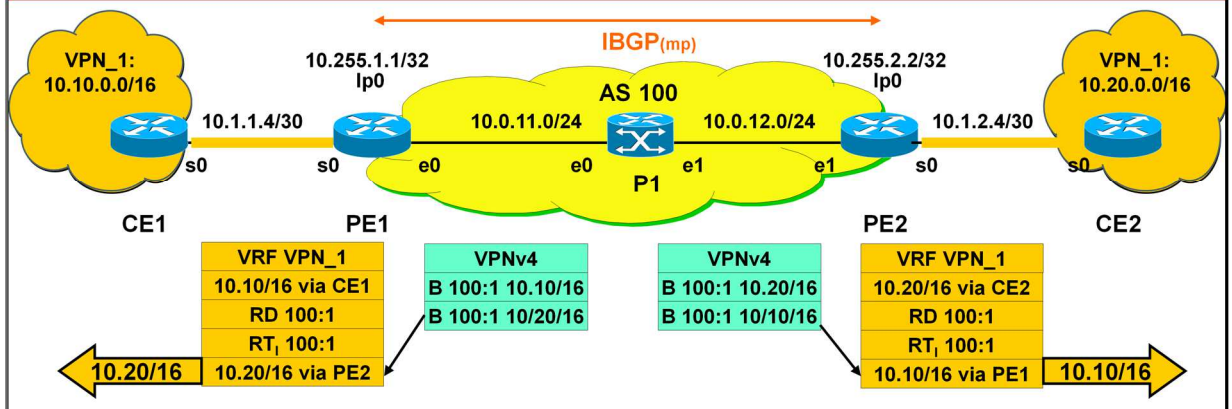
PE2:
ip vrf *VPN_1*
rd 100:1

router bgp 100
no bgp default ipv4-unicast
address-family ipv4 vrf *VPN_1*
redistribute rip
no auto-summary
no synchronization
exit-address-family

 MPLS-VPN v6.1 93

Appendix 4 - MPLS-VPN (v6.1)

Redistribution of VPNv4 routes into VRF RIPv2 (PE router)



VRF VPN_1
10.10/16 via CE1
RD 100:1
RT_I 100:1
10.20/16 via PE2

VPNv4
B 100:1 10.10/16
B 100:1 10/20/16

VPNv4
B 100:1 10.20/16
B 100:1 10/10/16

VRF VPN_1
10.20/16 via CE2
RD 100:1
RT_I 100:1
10.10/16 via PE1

10.20/16 — 10.10/16

PE1:

```
router rip
 version 2
 address-family ipv4 vrf VPN_1
   redistribute bgp 100 metric transparent
 exit-address-family

router bgp 100
  no bgp default ipv4-unicast
  address-family ipv4 vrf VPN_1
    redistribute rip
  exit-address-family
```

PE2:

```
router rip
 version 2
 address-family ipv4 vrf VPN_1
   redistribute bgp 100 metric transparent
 exit-address-family

router bgp 100
  no bgp default ipv4-unicast
  address-family ipv4 vrf VPN_1
     redistribute rip
  exit-address-family
```

 MPLS-VPN v6.1 94

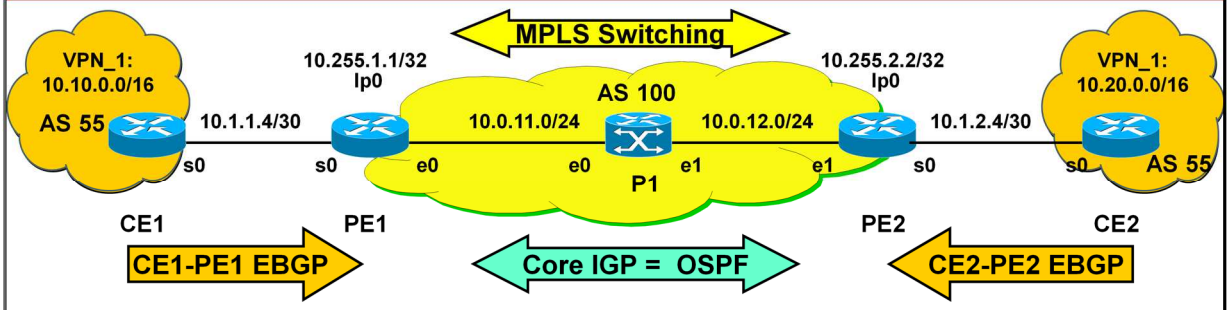
redistribute bgp 100 metric transparent
(preserves RIPv2 hops over MPLS-VPN)

Agenda

- MP-BGP
- VPN Overview
- MPLS VPN Architecture
- MPLS VPN Basic VPNs
- MPLS VPN Complex VPNs
- MPLS VPN Configuration (Cisco)
  - CE-PE OSPF Routing
  - CE-PE Static Routing
  - CE-PE RIP Routing
  - CE-PE External BGP Routing

Appendix 4 - MPLS-VPN (v6.1)

IP Addressing, EBGP Routing in VPN_1, Basic OSPF Routing and MPLS in AS 100



CE1:
```
int s0
  ip address 10.1.1.5 255.255.255.252
router bgp 55
  neighbor 10.1.1.6 remote-as 100
  network 10.10.0.0 mask 255.255.0.0
```

PE1:
```
ip cef
int loopback  0
  ip address 10.255.1.1 255.255.255.255
int s0
  ip address 10.1.1.6 255.255.255.252
int e0
  ip address 10.0.11.1 255.255.255.0
  mpls ip
router ospf 100
  network 10.0.11.1 0.0.0.0 area 0
```

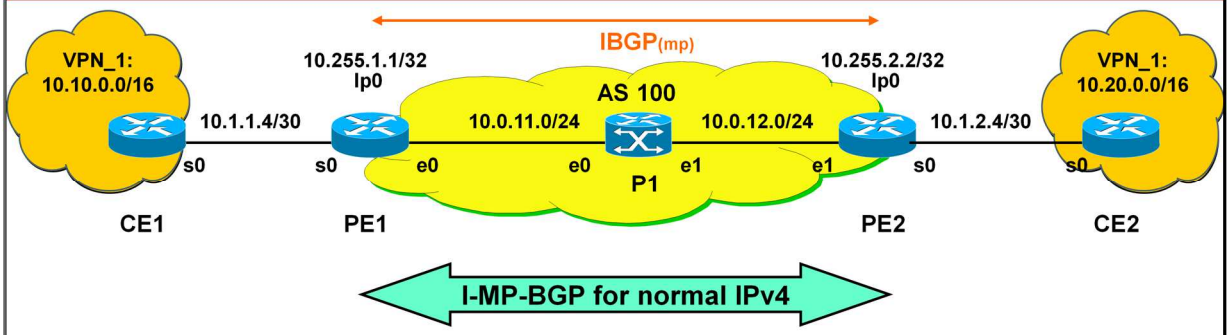
CE2:
```
int s0
  ip address 10.1.2.5 255.255.255.252
router bgp 55
  neighbor 10.1.2.6 remote-as 100
  network 10.20.0.0 mask 255.255.0.0
```

PE2:
```
ip cef
int loopback  0
  ip address 10.255.2.2 255.255.255.255
int s0
  ip address 10.1.2.6 255.255.255.252
int e0
  ip address 10.0.12.2 255.255.255.0
  mpls ip
router ospf 100
  network 10.0.12.1 0.0.0.0 area 0
```

 MPLS-VPN v6.1 96

Appendix 4 - MPLS-VPN (v6.1)

Start Normal I-BGP in AS 100



I-MP-BGP for normal IPv4

PE1:

```
int loopback  0
  ip address 10.255.1.1 255.255.255.255

router bgp 100
  no bgp default ipv4-unicast
  bgp router-id 10.255.1.1
  neighbor 10.255.2.2 remote-as 100
  neighbor 10.255.2.2 update-source loop 0
  address-family ipv4
    neighbor 10.255.2.2 next-hop-self
    neighbor 10.255.2.2 activate
    no auto-summary (default)
    no synchronization (default)
  exit address-family
```

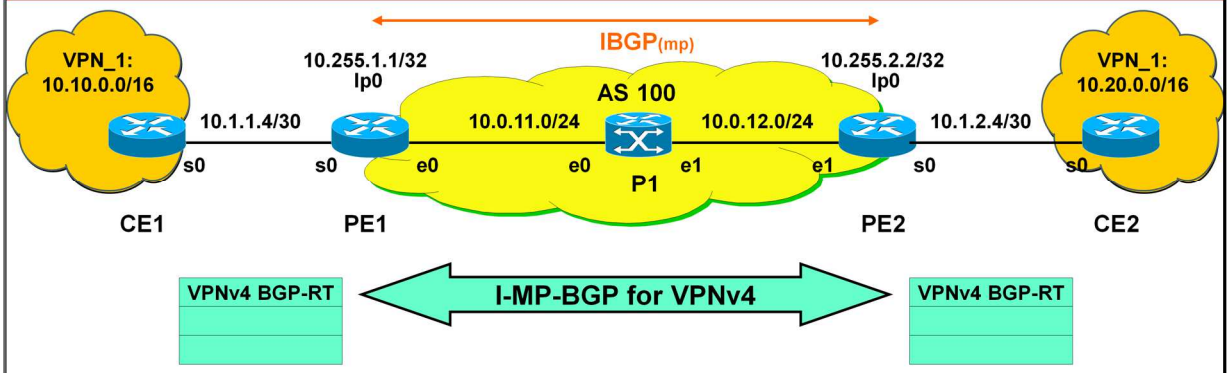
PE2:

```
int loopback  0
  ip address 10.255.2.2 255.255.255.255
router bgp 100
  no bgp default ipv4-unicast
  bgp router-id 10.255.2.2
  neighbor 10.255.1.1 remote-as 100
  neighbor 10.255.1.1 update-source loop 0
  address-family ipv4
    neighbor 10.255.1.1 next-hop-self
    neighbor 10.255.1.1 activate
    no auto-summary (default)
    no synchronization (default)
  exit address-family
```

 MPLS-VPN v6.1 97

Appendix 4 - MPLS-VPN (v6.1)

Start MP-BGP in AS 100



PE1:

```
int loopback  0
  ip address 10.255.1.1 255.255.255.255
router bgp 100
  no bgp default ipv4-unicast
  bgp router-id 10.255.1.1
  neighbor 10.255.2.2 remote-as 100
  neighbor 10.255.2.2 update-source loop 0
 address-family vpnv4
    neighbor 10.255.2.2 activate
    neighbor 10.255.2.2 next-hop-self
    neighbor 10.255.2.2 send-community extended (default)
  exit-address-family
```

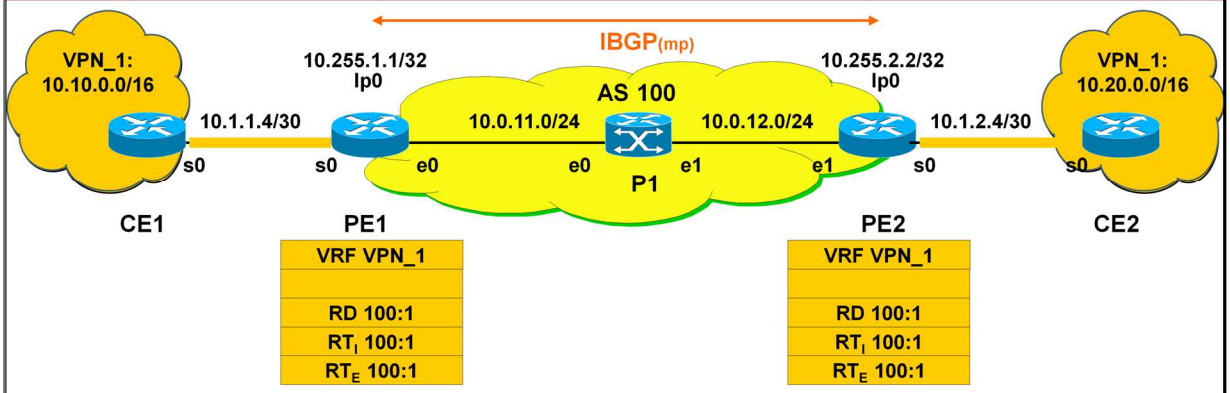
PE2:

```
int loopback  0
  ip address 10.255.2.2 255.255.255.255
router bgp 100
  no bgp default ipv4-unicast
  bgp router-id 10.255.2.2
  neighbor 10.255.1.1 remote-as 100
  neighbor 10.255.1.1 update-source loop 0
  address-family vpnv4
    neighbor 10.255.1.1 activate
    neighbor 10.255.1.1 next-hop-self
    neighbor 10.255.1.1 send-community extended
  exit-address-family
```

 MPLS-VPN v6.1 98

Appendix 4 - MPLS-VPN (v6.1)

Create VRF and Bring Interface into VRF (PE router)



VRF VPN_1
RD 100:1
RT_I 100:1
RT_E 100:1

VRF VPN_1
RD 100:1
RT_I 100:1
RT_E 100:1

PE1:

```
ip vrf VPN_1
  rd 100:1
  route-target import 100:1
  route-target export 100:1

int s0
  ip vrf forwarding VPN_1
  ip address 10.1.1.6 255.255.255.252
```

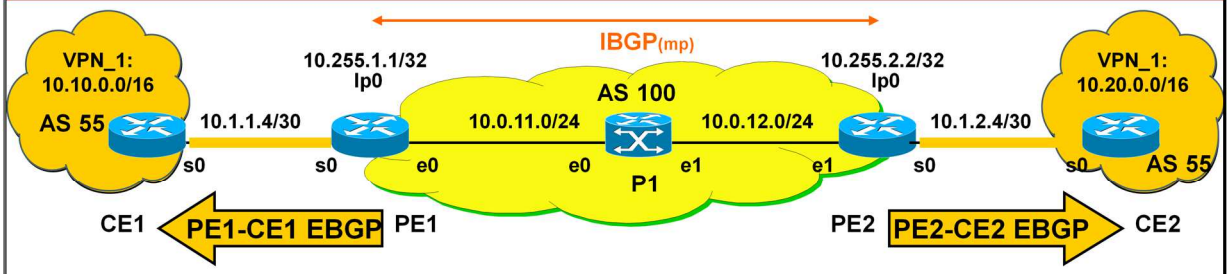
PE2:

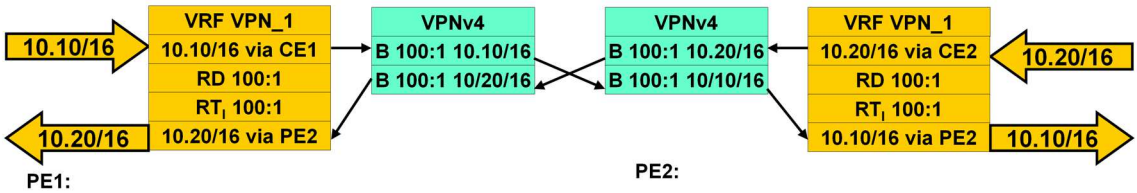
```
ip vrf VPN_1
  rd 100:1
  route-target import 100:1
  route-target export 100:1

int s0
  ip vrf forwarding VPN_1
  ip address 10.1.2.6 255.255.255.252
```

 MPLS-VPN v6.1 99

Appendix 4 - MPLS-VPN (v6.1)

Start Dynamic Routing towards CE using EBGP Redistribute into MP-BGP and vice versa





PE1:

```
ip vrf VPN_1
  rd 100:1

router bgp 100
  address-family ipv4 vrf VPN_1
    neighbor 10.1.1.5 remote-as 55
    neighbor 10.1.1.5 activate
    no auto-summary
    no synchronization
exit-address-family
```

PE2:

```
ip vrf VPN_1
  rd 100:1

router bgp 100
  address-family ipv4 vrf VPN_1
    neighbor 10.1.2.5 remote-as 55
    neighbor 10.1.2.5 activate
    no auto-summary
    no synchronization
exit-address-family
```

 MPLS-VPN v6.1 100